Intel Open Source Graphics Programmer's Reference Manual (PRM) for the 2013 Intel® Core™ Processor Family, including Intel HD Graphics, Intel Iris™ Graphics and Intel Iris Pro Graphics

Volume 11a: Display (Haswell)

12/18/2013

Copyright

Display

Table of Contents

VGA and Extended VGA Registers

This section describes the registers and the functional operation notations for the observable registers in the VGA section. This functionality is provided as a means for support of legacy applications and operating systems. It is important to note that these registers in general have the desired effects only when running VGA display modes.

The main exceptions to this are the palette interface which allows real mode DOS applications and full screen VGA applications under an OS control running in high resolution (non-VGA) modes to access the palette through the VGA register mechanisms and the use of the ST01 status bits that determine when the VGA enters display enable and sync periods. Other exceptions include the register bits that control the memory accesses through the A000:0000 and B000:0000 memory segments which are used during operating system emulation of VGA for "DOS box" applications. Some of the functions of the VGA are enabled or defeated through the programming of the VGA control register bits that are located in the MMIO register space.

Given the legacy nature of this function, it has been adapted to the changing environment that it must operate within. The three most notable changes are the addition of high resolution display mode support, new operating system support, and the use of fixed resolution display devices (such as LCD panels). Additional control bits in the PCI Config space will affect the ability to access the registers and memory aperture associated with VGA.

Mode of Operation	VGA Disable	VGA Display	VGA Registers	Palette (VGA)	VGA Memory	VGA Banking
VGA DOS	No	Yes	Yes	Yes	Yes	No
HiRes DOS	Yes	No	Yes	Yes	No	Yes
Fullscreen DOS	Yes/No	No/Yes	Yes	Yes	Yes	Yes
DOS Emulation	Yes	No	Yes	Yes	Yes	Yes

VGA Display Mode	Dot Clock Select	Dot Clock Range	132 Column Text Support	9-Dot Disable Support	Main Use
Native	VGA Clock Select	25/28 MHz	No	No	Analog CRT (VGA connector)
Centered	Fixed at display Requirements	Product Specific	No	Yes	Digital Display
Upper Left Corner	Fixed at display Requirements	Product Specific	No	Yes	Internal Panel

Native, Centered, and Upper Left Corner support varies from product to product.

Even in the native VGA display operational modes, not all combinations of bit settings result in functional operating modes. VGA display modes have the restriction that they can be used only when all other display planes are disabled.

These registers are accessed via I/O space. The I/O space resides in the PCI compatibility hole and uses only the addresses that were part of the original VGA I/O space (which includes EGA and MDA emulation). Accesses to the VGA I/O addresses are steered to the proper bus and rely on proper setup of bridge registers. Extended VGA registers such as GR10 and GR11 use additional indexes for the already defined I/O addresses. VGA register accesses are allowed as 8 or 16 bit naturally aligned transactions only. Word transactions must have the least significant bit of the address set to zero. DWORD I/O operations should not be performed on these registers.

Some products may support access to these registers through MMIO. The access method varies and is documented elsewhere.

General Control and Status Registers

The setup, enable, and general registers are all directly accessible by the CPU. A sub indexing scheme is not used to read from and write to these registers.

Name	Function	Read		Write	
		I/O	Memory Offset	I/O	Memory Offset
ST00	VGA Input Status Register 0	3C2h	3C2h	--	--
ST01	VGA Input Status Register 1	3BAh/3DAh[1]	3BAh/3DAh[1]	--	--
FCR	VGA Feature Control Register	3CAh	3CAh	3BAh/3DAh[1]	3BAh/3DAh[1]
MSR	VGA Miscellaneous Output Register	3CCh	3CCh	3C2h	3C2h

Note: [1] The address selection for ST01 reads and FCR writes is dependent on CGA or MDA emulation mode as selected via the MSR register.

Various bits in these registers provide control over the real-time status of the horizontal sync signal, the horizontal retrace interval, the vertical sync signal, and the vertical retrace interval. The horizontal retrace interval is the period during the drawing of each scan line containing active video data, when the active video data is not being displayed. This period includes the horizontal front and back porches, and the horizontal sync pulse. The horizontal retrace interval is always longer than the horizontal sync pulse. The vertical retrace interval is the period during which the scan lines not containing active video data are drawn. This includes the vertical front porch, back porch, and the vertical sync pulse. The vertical retrace interval is normally longer than the vertical sync pulse.

ST00 - Input Status 0

I/O (and Memory Offset) Address:3C2h

Default:00h

Attributes:Read Only

Bit	Descriptions
7	**CRT Interrupt Pending.** This bit is here for EGA compatibility and **will always return zero**. Note that the generation of interrupts was originally enabled, through bits [4,5] of the Vertical Retrace End Register (CR11). This ability to generate interrupts at the start of the vertical retrace interval is a feature that is typically unused by DOS software and therefore is only supported through other means for use under a operating system support. 0 = CRT (vertical retrace interval) interrupt is not pending. 1 = CRT (vertical retrace interval) interrupt is pending
6:5	**Reserved.** Read as 0s.
4	**RGB Comparator / Sense.** This bit is here for compatibility and **will always return one**. Monitor detection must be done be done through the programming of registers in the MMIO space. 0 = Below threshold 1 = Above threshold
3:0	**Reserved.** Read as 0s.

ST01 - Input Status 1

I/O (and Memory Offset) Address:3BAh/3DAh

Default:00h

Attributes:Read Only

The address selection is dependent on CGA or MDA emulation mode as selected via the MSR register.

Bit	Descriptions
7	**Reserved (as per VGA specification).** Read as 0s.
6	**Reserved.** Read as 0.
5:4	**Video Feedback 1, 0.** These bits are connected to 2 of the 8 color bits sent to the palette. Bits 4 and 5 of the Color Plane Enable Register (AR12) selects which two of the 8 possible color bits become connected to these 2 bits of this register. These bits exist for EGA compatibility.
3	**Vertical Retrace/Video.** 0 = VSYNC inactive (Indicates that a vertical retrace interval is not taking place). 1 = VSYNC active (Indicates that a vertical retrace interval is taking place). **Note:** VGA pixel generation is not locked to the display output but is loosely coupled. A VSYNC indication may not occur during the actual VSYNC going to the display but during the VSYNC that is generated as part of the VGA pixel generation. The exact relationship will vary with the VGA display operational mode. This status bit will remain active when the VGA is disabled and the device is running in high resolution modes (non-VGA) to allow for applications that (now incorrectly) use these status registers bits. In this case, the status will come from the pipe that the VGA is assigned to. Bits 4 and 5 of the Vertical Retrace End Register (CR11) previously could program this bit to generate an interrupt at the start of the vertical retrace interval. This ability to generate interrupts at the start of the vertical retrace interval is a feature that is largely unused by legacy software. Interrupts are not supported through the VGA register bits.
2:1	**Reserved.** Read as 0s.

Bit	Descriptions
0	**Display Enable Output.** Display Enable is a status bit (bit 0) in VGA Input Status Register 1 that indicates when either a horizontal retrace interval or a vertical retrace interval is taking place. This bit was used with the EGA graphics system (and the ones that preceded it, including MDA and CGA). In those cases, it was important to check the status of this bit to ensure that one or the other retrace intervals was taking place before reading from or writing to the frame buffer. In these earlier systems, reading from or writing to the frame buffer at times outside the retrace intervals meant that the CRT controller would be denied access to the frame buffer. Those behaviors resulted in either "snow" or a flickering display. This bit provides compatibility with software designed for those early graphics controllers. This bit is currently used in DOS applications that access the palette to prevent the sparkle associated with read and write accesses to the palette RAM with the same address on the same clock cycle. **This status bit remains active when the VGA display is disabled and the device is running in high resolution modes (non-VGA) to allow for applications that (now considered incorrect) use these status registers bits. In this case, the status will come from the pipe that the VGA is assigned to. When in panel fitting VGA or centered VGA operation, the meaning of these bits will not be consistent with native VGA timings.** 0 = Active display data is being sent to the display. Neither a horizontal retrace interval or a vertical retrace interval is currently taking place. 1 = Either a horizontal retrace interval (horizontal blanking) or a vertical retrace interval (vertical blanking) is currently taking place.

FCR - Feature Control

I/O (and Memory Offset) Address:3BAh/3DAh - Write; 3CAh - Read

Default:00h

Attributes:See Address above

The I/O address used for writes is dependent on CGA or MDA emulation mode as selected via the MSR register. In the original EGA, bits 0 and 1 were used as part of the feature connector interface. Feature connector is not supported in these devices and those bits will always read as zero.

Bit	Descriptions
7:4	**Reserved.** Read as 0.
3	**VSYNC Control.** This bit is provided for compatibility only and has no other function. Reads and writes to this bit have no effect other than to change the value of this bit. The previous definition of this bit selected the output on the VSYNC pin. 0 = Was used to set VSYNC output on the VSYNC pin (default). 1 = Was used to set the logical 'OR' of VSYNC and Display Enable output on the VSYNC pin. This capability was not typically very useful.
2:0	**Reserved.** Read as 0.

MSR - Miscellaneous Output

I/O (and Memory Offset) Address:3C2h - Write; 3CCh - Read

Default:00h

Attributes:See Address above

Bit	Descriptions
7	**CRT VSYNC Polarity.** This is a legacy function that is used in native VGA modes. For most cases, sync polarity will be controlled by the port control bits. The VGA settings can be optionally selected for compatibility with the original VGA when used in the VGA native mode. Sync polarity was used in VGA to signal the monitor how many lines of active display are being generated. 0 = Positive Polarity (default). 1 = Negative Polarity.
6	**CRT HSYNC Polarity.** This is a legacy function that is used in native VGA modes. For most cases, sync polarity will be controlled by the port control bits. The VGA settings can be optionally selected for compatibility with the original VGA when used in the VGA native mode. 0 = Positive Polarity (default). 1 = Negative Polarity
5	**Page Select.** In Odd/Even Memory Map Mode 1 (GR6), this bit selects the upper or lower 64 KB page in display memory for CPU access: 0 = Upper page (default) 1 = Lower page. Selects between two 64KB pages of frame buffer memory during standard VGA odd/even modes (modes 0h through 5h). Bit 1 of register GR06 can also program this bit in other modes. Note that this bit is would normally set to 1 by the software.
4	**Reserved.** Read as 0.

Bit	Descriptions
3:2	**Clock Select.** These bits can select the dot clock source for the CRT interface. The bits should be used to select the dot clock in standard native VGA modes only. When in the centering or upper left corner modes, these bits should be set to have no effect on the clock rate. The actual frequencies that these bits select, if they have any affect at all, is programmable through the DPLL MMIO registers. 00 = CLK0, 25.175 MHz (for standard VGA modes with 640 pixel (8-dot) horizontal resolution) (default) 01 = CLK1, 28.322 MHz. (for standard VGA modes with 720 pixel (9-dot) horizontal resolution) 10 = Was used to select an external clock (now unused) 11 = Reserved
1	**A0000-BFFFFh Memory Access Enable.** VGA Compatibility bit enables access to video memory (frame buffer) at A0000-BFFFFh. When disabled, accesses to VGA memory are blocked in this region. This bit is independent of and does not block CPU access to the video linear frame buffer at other addresses. 0 = Prevent CPU access to memory/registers/ROM through the A0000-BFFFF VGA memory aperture (default). 1 = Allow CPU access to memory/registers/ROM through the A0000-BFFFF VGA memory aperture. This memory must be mapped as UC by the CPU.
0	**I/O Address Select.** This bit selects 3Bxh or 3Dxh as the I/O address for the CRT Controller registers, the Feature Control Register (FCR), and Input Status Register 1 (ST01). Presently ignored (whole range is claimed), but will "ignore" 3Bx for color configuration or 3Dx for monochrome. Note that it is typical in AGP chipsets to shadow this bit and properly steer I/O cycles to the proper bus for operation where a MDA exists on another bus such as ISA. 0 = Select 3Bxh I/O address (MDA emulation) (default). 1 = Select 3Dxh I/O address (CGA emulation).

Note: In standard VGA modes using the analog VGA connector, bits 7 and 6 indicate which of the three standard VGA vertical resolutions the standard VGA display should use. Extended modes, including those with a vertical resolution of 480 scan lines, may use a setting of 0 for both of these bits. Different connector standards and timing standards specify the proper use of sync polarity. This setting was "reserved" in the VGA standard.

Analog CRT Display Sync Polarities

V	H	Display	Horizontal Frequency	Vertical Frequency
P	P	200 Line	15.7 KHz	60 Hz
N	P	350 Line	21.8 KHz	60 Hz
P	N	400 Line	31.5 KHz	70 Hz
N	N	480 Line	31.5 KHz	60 Hz

Sequencer Registers

The sequencer registers are accessed via either I/O or Memory. To access registers the VGA Sequencer Index register (SRX) at I/O address 3C4h (or memory address 3C4h) is written with the index of the desired register. Then the desired register is accessed through the data port for the sequencer registers at I/O address 3C5 (or memory address 3C5).

SRX - Sequencer Index

I/O (and Memory Offset) Address:3C4h

Default:00h

Attributes:Read/Write

Bit	Description
7:3	**Reserved.** Read as 0s.
2:0	**Sequencer Index.** This field contains a 3-bit Sequencer Index value used to access sequencer data registers at indices 0 through 7.

SR00 - Sequencer Reset

I/O (and Memory Offset) Address:3C5h(Index=00h)

Default:00h

Attributes:Read/Write

Bit	Descriptions
7:2	**Reserved.** Read as 0.
1	**Reserved.** Reserved for VGA compatibility (was reset).
0	**Reserved.** Reserved for VGA compatibility. (was reset)

SR01 - Clocking Mode

I/O (and Memory Offset) Address:3C5h (Index=01h)

Default:00h

Attributes:Read/Write

Bit	Descriptions
7:6	**Reserved.** Read as 0s.
5	**Screen Off.** 0 = Normal Operation (default). 1 = Disables video output (blanks the screen) and turns off display data fetches. Synchronization pulses to the display, however, are maintained. Setting this bit to 1 had been used as a way to more rapidly update and improve CPU access performance to the frame buffer during VGA modes. In non-VGA modes (VGA Disable=1), this bit has no effect. Before the VGA is disabled through the MMIO VGA control register, this bit should be set to stop the memory accesses from the display. **Programming Notes:** The following sequence must be used when disabling the VGA plane. 1. Write SR01 to set bit 5 = 1 to disable video output. 2. Wait for 100us. 3. Disable the VGA plane via Bit 31 of the MMIO VGA control register (location found in the MMIO display register programming specification).
4	**Shift 4.** 0 = Load video shift registers every 1 or 2 character clocks (depending on bit 2 of this register) (default). 1 = Load shift registers every 4th character clock.
3	**Dot Clock Divide.** Setting this bit to 1 stretches doubles all horizontal timing periods that are specified in the VGA horizontal CRTC registers. This bit is used in standard VGA 40-column text modes to stretch timings to create horizontal resolutions of either 320 or 360 pixels (as opposed to 640 or 720 pixels, normally used in standard VGA 80-column text modes). The effect of this is that there will actually be twice the number of pixels sent to the display per line. 0 = Pixel clock is left unaltered (used for 640 (720) pixel modes); (default). 1 = Pixel clock divided by 2 (used for 320 (360) pixel modes).

Bit	Descriptions
2	**Shift Load.** Bit 4 of this register must be 0 for this bit to be effective. 0 = Load video data shift registers every character clock (default). 1 = Load video data shift registers every other character clock.
1	**Reserved.** Read as 0.
0	**8/9 Dot Clocks.** This bit determines whether a character clock is 8 or 9 dot clocks long if clock doubling is disabled and 16 or 18 clocks if it is. This also changes the interpretation of the pixel panning values (see chart). An additional control bit determines if this bit is to be ignored and 8-dot characters are to be used always. The 9-dot disable would be used when doubling the horizontal pixels on a 1280 wide display or non-doubling on a 640 wide display. Panning however will occur according to the expected outcome. 0 = 9 dot clocks (9 horizontal pixels) per character in text modes with a horizontal resolution of 720 pixels. 1 = 8 dot clocks (8 horizontal pixels) per character in text or graphics modes with a horizontal resolution of 640 pixels.

SR02 - Plane/Map Mask

I/O (and Memory Offset) Address:3C5h (Index=02h)

Default:00h

Attributes:Read/Write

Bit	Descriptions
7:4	**Reserved.** Read as 0s.
3:0	**Memory Planes [3:0] Processor Write Access Enable.** In both the Odd/Even Mode and the Chain 4 Mode, these bits still control access to the corresponding color plane. 0 = Disable. 1 = Enable. **Note:**This register is referred to in the VGA standard as the Map Mask Register.

SR03 - Character Font

I/O (and Memory Offset) Address:3C5h (index=03h)

Default:00h

Attributes:Read/Write

<table>
<tr><th>Bit</th><th>Descriptions</th></tr>
<tr><td>7:6</td><td>Reserved. Read as 0s.</td></tr>
<tr><td>3:2,5</td><td>Character Map Select Bits for Character Map B. These three bits are used to select the character map (character generator tables) to be used as the secondary character set (font). Note that the numbering of the maps is not sequential.
<table>
<tr><th>Bit [3:2, 5]</th><th>Map Number</th><th>Table Location</th></tr>
<tr><td>00,0</td><td>0</td><td>1st 8KB of plane 2 at offset 0 (default)</td></tr>
<tr><td>00,1</td><td>4</td><td>2nd 8KB of plane 2 at offset 8K</td></tr>
<tr><td>01,0</td><td>1</td><td>3rd 8KB of plane 2 at offset 16K</td></tr>
<tr><td>01,1</td><td>5</td><td>4th 8KB of plane 2 at offset 24K</td></tr>
<tr><td>10,0</td><td>2</td><td>5th 8KB of plane 2 at offset 32K</td></tr>
<tr><td>10,1</td><td>6</td><td>6th 8KB of plane 2 at offset 40K</td></tr>
<tr><td>11,0</td><td>3</td><td>7th 8KB of plane 2 at offset 48K</td></tr>
<tr><td>11,1</td><td>7</td><td>8th 8KB of plane 2 at offset 56K</td></tr>
</table></td></tr>
<tr><td>1:0,4</td><td>Character Map Select Bits for Character Map A. These three bits are used to select the character map (character generator tables) to be used as the primary character set (font). Note that the numbering of the maps is not sequential.
<table>
<tr><th>Bit [1:0,4]</th><th>Map Number</th><th>Table Location</th></tr>
<tr><td>00,0</td><td>0</td><td>1st 8KB of plane 2 at offset 0 (default)</td></tr>
<tr><td>00,1</td><td>4</td><td>2nd 8KB of plane 2 at offset 8K</td></tr>
<tr><td>01,0</td><td>1</td><td>3rd 8KB of plane 2 at offset 16K</td></tr>
<tr><td>01,1</td><td>5</td><td>4th 8KB of plane 2 at offset 24K</td></tr>
<tr><td>10,0</td><td>2</td><td>5th 8KB of plane 2 at offset 32K</td></tr>
<tr><td>10,1</td><td>6</td><td>6th 8KB of plane 2 at offset 40K</td></tr>
<tr><td>11,0</td><td>3</td><td>7th 8KB of plane 2 at offset 48K</td></tr>
<tr><td>11,1</td><td>7</td><td>8th 8KB of plane 2 at offset 56K</td></tr>
</table></td></tr>
</table>

NOTES:

1. In text modes, bit 3 of the video data's attribute byte normally controls the foreground intensity. This bit may be redefined to control switching between character sets. This latter function is enabled whenever there is a difference in the values of the Character Font Select A and the Character Font Select B bits. If the two values are the same, the character select function is disabled and attribute bit 3 controls the foreground intensity.
2. Bit 1 of the Memory Mode Register (SR04) must be set to 1 for the character font select function of this register to be active. Otherwise, only character maps 0 and 4 are available.

SR04 - Memory Mode Register

I/O (and Memory Offset) Address:3C5h (index=04h)

Default:00h

Attributes:Read/Write

Bit	Description
7:4	**Reserved.** Read as 0.
3	**Chain 4 Mode.** The selections made by this bit affect both CPU Read and write accesses to the frame buffer. 0 = The manner in which the frame buffer memory is mapped is determined by the setting of bit 2 of this register (default). 1 = The frame buffer memory is mapped in such a way that the function of address bits 0 and 1 are altered so that they select planes 0 through 3. This setting is used in mode x13 to allow all four planes to be accessed via sequential addresses.
2	**Odd/Even Mode.** Bit 3 of this register must be set to 0 for this bit to be effective. The selections made by this bit affect only non-paged CPU accesses to the frame buffer through the VGA aperture. 0 = The frame buffer memory is mapped in such a way that the function of address bit 0 such that even addresses select planes 0 and 2 and odd addresses select planes 1 and 3 (default). 1 = Addresses sequentially access data within a bit map, and the choice of which map is accessed is made according to the value of the Plane Mask Register (SR02).
1	**Extended Memory Enable.** This bit must be set to 1 to enable the selection and use of character maps in plane 2 via the Character Map Select Register (SR03). 0 = Disable CPU accesses to more than the first 64KB of VGA standard memory (default). 1 = Enable CPU accesses to the rest of the 256KB total VGA memory beyond the first 64KB.
0	**Reserved.** Read as 0.

SR07 - Horizontal Character Counter Reset

I/O (and Memory Offset) Address:3C5h (index=07h)

Default:00h

Attributes:Read/Write

For standard VGAs, writing this register (with any data) causes the horizontal character counter to be held in reset (the character counter output will remain 0). It remained in reset until a write occurred to any other sequencer register location with SRX set to an index of 0 through 6. In this implementation that sequence has no such special effect.

The vertical line counter is clocked by a signal derived from the horizontal display enable (which does not occur if the horizontal counter is held in reset). Therefore, if a write occurs to this register during the vertical retrace interval, both the horizontal and vertical counters will be set to 0. A write to any other sequencer register location (with SRX set to an index of 0 through 6) may then be used to start both counters with reasonable synchronization to an external event via software control. Although this was a standard VGA register, it was not documented.

Bit	Description
7:0	**Horizontal Character Counter.**

Graphics Controller Registers

The graphics controller registers are accessed via either I/O or Memory. Accesses to the registers of the VGA Graphics Controller are done through the use of address 3CEh (or memory address 3CEh) written with the index of the desired register. Then the desired register is accessed through the data port for the graphics controller registers at I/O address 3CFh (or memory address 3CFh). Indexes 10 and 11 should only be accessed through the I/O space only.

GRX - GRX Graphics Controller Index Register

I/O (and Memory Offset) Address:3CEh

Default:000UUUUUb (U=Undefined)

Attributes:Read/Write

Bit	Description
7:5	**Reserved.** Read as 0.
4:0	**Graphics Controller Register Index.** This field selects any one of the graphics controller registers (GR00-GR18) to be accessed via the data port at I/O (or memory offset) location 3CFh.

GR00 - Set/Reset Register

I/O (and Memory Offset) Address:3CFh (index=00h)

Default:0Uh (U=Undefined)

Attributes:Read/Write

Bit	Description
7:4	**Reserved.** Read as 0.
3:0	**Set/Reset Plane [3:0].** When the Write Mode bits (bits 0 and 1) of the Graphics Mode Register (GR05) are set to select Write Mode 0, all 8 bits of each byte of each memory plane are set to either 1 or 0 as specified in the corresponding bit in this register, if the corresponding bit in the Enable Set/Reset Register (GR01) is set to 1. When the Write Mode bits (bits 0 and 1) of the Graphics Mode Register (GR05) are set to select Write Mode 3, all CPU data written to the frame buffer is rotated, then logically ANDed with the contents of the Bit Mask Register (GR08), and then treated as the addressed data's bit mask, while value of these four bits of this register are treated as the color value.

GR01 - Enable Set/Reset Register

I/O (and Memory Offset) Address:3CFh (Index=01h)

Default:0Uh (U=Undefined)

Attributes:Read/Write

Bit	Description
7:4	**Reserved.** Read as 0.
3:0	**Enable Set/Reset Plane [3:0].** This register works in conjunction with the Set/Reset Register (GR00). The Write Mode bits (bits 0 and 1) must be set for Write Mode 0 for this register to have any effect. 0 = The corresponding memory plane can be read from or written to by the CPU without any special bitwise operations taking place. 1 = The corresponding memory plane is set to 0 or 1 as specified in the Set/Reset Register (GR00).

GR02 - Color Compare Register

I/O (and Memory Offset) Address:3CFh (Index=02h)

Default:0Uh (U=Undefined)

Attributes:Read/Write

Bit	Description
7:4	**Reserved.** Read as 0.
3:0	**Color Compare Plane [3:0].** When the Read Mode bit (bit 3) of the Graphics Mode Register (GR05) is set to select Read Mode 1, all 8 bits of each byte of each of the 4 memory planes of the frame buffer corresponding to the address from which a CPU read access is being performed are compared to the corresponding bits in this register (if the corresponding bit in the Color Don't Care Register (GR07) is set to 1). The value that the CPU receives from the read access is an 8-bit value that shows the result of this comparison, wherein value of 1 in a given bit position indicates that all of the corresponding bits in the bytes across all of the memory planes that were included in the comparison had the same value as their memory plane's respective bits in this register.

GR03 - Data Rotate Register

I/O (and Memory Offset) Address:3CFh (Index=03h)

Default:0Uh (U=Undefined)

Attributes:Read/Write

Bit	Description
7:5	**Reserved.** Read as 0.
4:3	**Function Select.** These bits specify the logical function (if any) to be performed on data that is meant to be written to the frame buffer (using the contents of the memory read latch) just before it is actually stored in the frame buffer at the intended address location. 00 = Data being written to the frame buffer remains unchanged, and is simply stored in the frame buffer. 01 = Data being written to the frame buffer is logically ANDed with the data in the memory read latch before it is actually stored in the frame buffer. 10 = Data being written to the frame buffer is logically ORed with the data in the memory read latch before it is actually stored in the frame buffer. 11 = Data being written to the frame buffer is logically XORed with the data in the memory read latch before it is actually stored in the frame buffer.
2:0	**Rotate Count.** These bits specify the number of bits to the right to rotate any data that is meant to be written to the frame buffer just before it is actually stored in the frame buffer at the intended address location.

GR04 - Read Plane Select Register

I/O (and Memory Offset) Address:3CFh (Index=04h)

Default:0Uh (U=Undefined)

Attributes:Read/Write

Bit	Description
7:2	**Reserved.** Read as 0.
1:0	**Read Plane Select.** These two bits select the memory plane from which the CPU reads data in Read Mode 0. In Odd/Even Mode, bit 0 of this register is ignored. In Chain 4 Mode, both bits 1 and 0 of this register are ignored. The four memory planes are selected as follows: 00 = Plane 0 01 = Plane 1 10 = Plane 2 11 = Plane 3 These two bits also select which of the four memory read latches may be read via the Memory read Latch Data Register (CR22). The choice of memory read latch corresponds to the choice of plane specified in the table above. The Memory Read Latch Data register and this additional function served by 2 bits are features of the VGA standard that were never documented.

GR05 - Graphics Mode Register

I/O (and Memory Offset) Address:3CFh (Index=05h)

Default:0UUU U0UUb (U=Undefined)

Attributes:Read/Write

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>Reserved. Read as 0.</td></tr>
<tr><td>6:5</td><td>Shift Register Control. In standard VGA modes, pixel data is transferred from the 4 graphics memory planes to the palette via a set of 4 serial output bits. These 2 bits of this register control the format in which data in the 4 memory planes is serialized for these transfers to the palette.

Bits [6:5]=00

One bit of data at a time from parallel bytes in each of the 4 memory planes is transferred to the palette via the 4 serial output bits, with 1 of each of the serial output bits corresponding to a memory plane. This provides a 4-bit value on each transfer for 1 pixel, making possible a choice of 1 of 16 colors per pixel.

Serial

<table>
<tr><th>Out</th><th>1st Xfer</th><th>2nd Xfer</th><th>3rd Xfer</th><th>4th Xfer</th><th>5th Xfer</th><th>6th Xfer</th><th>7th Xfer</th><th>8th Xfer</th></tr>
<tr><td>Bit 3</td><td>plane3 bit7</td><td>plane3 bit6</td><td>plane3 bit5</td><td>plane3 bit4</td><td>plane3 bit3</td><td>plane3 bit2</td><td>plane3 bit1</td><td>plane3 bit0</td></tr>
<tr><td>Bit 2</td><td>plane2 bit7</td><td>plane2 bit6</td><td>plane2 bit5</td><td>plane2 bit4</td><td>plane2 bit3</td><td>plane2 bit2</td><td>plane2 bit1</td><td>plane2 bit0</td></tr>
<tr><td>Bit 1</td><td>plane1 bit7</td><td>plane1 bit6</td><td>plane1 bit5</td><td>plane1 bit4</td><td>plane1 bit3</td><td>plane1 bit2</td><td>plane1 bit1</td><td>plane1 bit0</td></tr>
<tr><td>Bit 0</td><td>plane0 bit7</td><td>plane0 bit6</td><td>plane0 bit5</td><td>plane0 bit4</td><td>plane0 bit3</td><td>plane0 bit2</td><td>plane0 bit1</td><td>plane0 bit0</td></tr>
</table>

Bits [6:5]=01

Two bits of data at a time from parallel bytes in each of the 4 memory planes are transferred to the palette in a pattern that alternates per byte between memory planes 0 and 2, and memory planes 1 and 3. First the even-numbered and odd-numbered bits of a byte in memory plane 0 are transferred via serial output bits 0 and 1, respectively, while the even-numbered and odd-numbered bits of a byte in memory plane 2 are transferred via serial output bits 2 and 3. Next, the even-numbered and odd-numbered bits of a byte in memory plane 1 are transferred via serial output bits 0 and 1, respectively, while the even-numbered and odd-numbered bits of memory plane 3 are transferred via serial out bits 1 and 3. This provides a pair of 2-bit values (one 2-bit value for each of 2 pixels) on each transfer, making possible a choice of 1 of 4 colors per pixel.</td></tr>
</table>

Bit	Description
	Serial Out 1st Xfer 2nd Xfer 3rd Xfer 4th Xfer 5th Xfer 6th Xfer 7th Xfer 8th Xfer Bit 3 plane2 bit7 plane2 bit5 plane2 bit3 plane2 bit1 plane3 bi7t plane3 bit5 plane3 bit3 plane3 bit1 Bit 2 plane2 bit6 plane2 bit4 plane2 bit2 plane2 bit0 plane3 bit6 plane3 bit4 plane3 bit2 plane3 bit0 Bit 1 plane0 bit7 plane0 bit5 plane0 bit3 plane0 bit1 plane1 bit7 plane1 bit5 plane1 bit3 plane1 bit1 Bit 0 plane0 bit6 plane0 bit4 plane0 bit2 plane0 bit0 plane1 bit6 plane1 bit4 plane1 bit2 plane1 bit0 This alternating pattern is meant to accommodate the use of the Odd/Even mode of organizing the 4 memory planes, which is used by standard VGA modes 2h and 3h. **Bits [6:5]=1x** Four bits of data at a time from parallel bytes in each of the 4 memory planes are transferred to the palette in a pattern that iterates per byte through memory planes 0 through 3. First the 4 most significant bits of a byte in memory plane 0 are transferred via the 4 serial output bits, followed by the 4 least significant bits of the same byte. Next, the same transfers occur from the parallel byte in memory planes 1, 2 and lastly, 3. Each transfer provides either the upper or lower half of an 8 bit value for the color for each pixel, making possible a choice of 1 of 256 colors per pixel. This is the setting used in mode x13. Serial Out 1st Xfer 2nd Xfer 3rd Xfer 4th Xfer 5th Xfer 6th Xfer 7th Xfer 8th Xfer Bit 3 plane0 bit7 plane0 bit3 plane1 bit7 plane1 bit3 plane2 bit7 plane2 bit3 plane3 bit7 plane3 b3it Bit 2 plane0 bit6 plane0 bit2 plane1 bit6 plane1 bit2 plane2 bit6 plane2 bit2 plane3 bit6 plane3 bit2 Bit 1 plane0 bit5 plane0 bit1 plane1 bit5 plane1 bit1 plane2 bit5 plane2 bit1 plane3 bit5 plane3 bit1 Bit 0 plane0 bit4 plane0 bit0 plane1 bit4 plane1 bit0 plane2 bit4 plane2 bit0 plane3 bit4 plane3 bit0 This pattern is meant to accommodate mode 13h, a standard VGA 256-color graphics mode.

Bit	Description
4	**Odd/Even Mode.** 0 = Addresses sequentially access data within a bit map, and the choice of which map is accessed is made according to the value of the Plane Mask Register (SR02). 1 = The frame buffer is mapped in such a way that the function of address bit 0 is such that even addresses select memory planes 0 and 2 and odd addresses select memory planes 1 and 3. **Note:**This works in a way that is the inverse of (and is normally set to be the opposite of) bit 2 of the Memory Mode Register (SR02).
3	**Read Mode.** 0 = During a CPU read from the frame buffer, the value returned to the CPU is data from the memory plane selected by bits 1 and 0 of the Read Plane Select Register (GR04). 1 = During a CPU read from the frame buffer, all 8 bits of the byte in each of the 4 memory planes corresponding to the address from which a CPU read access is being performed are compared to the corresponding bits in this register (if the corresponding bit in the Color Don't Care Register (GR07) is set to 1). The value that the CPU receives from the read access is an 8-bit value that shows the result of this comparison. A value of 1 in a given bit position indicates that all of the corresponding bits in the bytes across all 4 of the memory planes that were included in the comparison had the same value as their memory plane's respective bits in this register.
2	**Reserved.** Read as 0.

Bit	Description
1:0	**Write Mode.** 00 = Write Mode 0 - During a CPU write to the frame buffer, the addressed byte in each of the 4 memory planes is written with the CPU write data after it has been rotated by the number of counts specified in the Data Rotate Register (GR03). If, however, the bit(s) in the Enable Set/Reset Register (GR01) corresponding to one or more of the memory planes is set to 1, then those memory planes will be written to with the data stored in the corresponding bits in the Set/Reset Register (GR00). 01 = Write Mode 1 - During a CPU write to the frame buffer, the addressed byte in each of the 4 memory planes is written to with the data stored in the memory read latches. (The memory read latches stores an unaltered copy of the data last read from any location in the frame buffer.) 10 = Write Mode 2 - During a CPU write to the frame buffer, the least significant 4 data bits of the CPU write data is treated as the color value for the pixels in the addressed byte in all 4 memory planes. The 8 bits of the Bit Mask Register (GR08) are used to selectively enable or disable the ability to write to the corresponding bit in each of the 4 memory planes that correspond to a given pixel. A setting of 0 in a bit in the Bit Mask Register at a given bit position causes the bits in the corresponding bit positions in the addressed byte in all 4 memory planes to be written with value of their counterparts in the memory read latches. A setting of 1 in a Bit Mask Register at a given bit position causes the bits in the corresponding bit positions in the addressed byte in all 4 memory planes to be written with the 4 bits taken from the CPU write data to thereby cause the pixel corresponding to these bits to be set to the color value. 11 = Write Mode 3 - During a CPU write to the frame buffer, the CPU write data is logically ANDed with the contents of the Bit Mask Register (GR08). The result of this ANDing is treated as the bit mask used in writing the contents of the Set/Reset Register (GR00) are written to addressed byte in all 4 memory planes.

GR06 - Miscellaneous Register

I/O (and Memory Offset) Address:3CFh (Index=06h)

Default: 0Uh (U=Undefined)

Attributes:Read/Write

Bit	Description
7:4	**Reserved.** Read as 0s.
3:2	**Memory Map Mode.** These 2 bits control the mapping of the VGA address range for frame buffer into the CPU address space as follows: 00 = A0000h - BFFFFh 01 = A0000h - AFFFFh 10 = B0000h - B7FFFh 11 = B8000h - BFFFFh Note: • This function is used in standard VGA modes, extended VGA modes (132 column text), and in non-VGA modes (hi-res). 132 column text modes are no longer supported. • VGA aperture memory accesses are also controlled by the PCI configuration Memory Enable bit and MSR<1>. • For accesses using GR10 and GR11 to paged VGA RAM or to device MMIO registers, set these bits to 01 to select the (A0000-AFFFF) range. • The CPU must map this memory as uncacheable (UC).
1	**Chain Odd/Even.** This bit provides the ability to alter the interpretation of address bit A0, so that it may be used in selecting between the odd-numbered memory planes (planes 1 and 3) and the even-numbered memory planes (planes 0 and 2). 0 = A0 functions normally. 1 = A0 is switched with a high order address bit, in terms of how it is used in address decoding. The result is that A0 is used to determine which memory plane is being accessed (A0=0 for planes 0 and 2 and A0=1 for planes 1 and 3).

Bit	Description
0	**Graphics/Text Mode.** This is one of two bits that are used to determine if the VGA is operating in text or graphics modes. The other bit is in AR10[0], these two bits need to be programmed in a consistent manner to achieve the proper results. 0 = Text mode. 1 = Graphics mode.

GR07 - Color Don't Care Register

I/O (and Memory Offset) Address:3CFh (Index=07h)

Default: 0Uh (U=Undefined)

Attributes:Read/Write

Bit	Description
7:4	**Reserved.** Read as 0.
3:0	**Ignore Color Plane [3:0].** Note that these bits have effect only when bit 3 of the Graphics Mode Register (GR05) is set to 1 to select read mode 1. 0 = The corresponding bit in the Color Compare Register (GR02) will not be included in color comparisons. 1 = The corresponding bit in the Color Compare Register (GR02) is used in color comparisons.

GR08 - Bit Mask Register

I/O (and Memory Offset) Address:3CFh (Index=08h)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Bit Mask.** 0 = The corresponding bit in each of the 4 memory planes is written to with the corresponding bit in the memory read latches. 1 = Manipulation of the corresponding bit in each of the 4 memory planes via other mechanisms is enabled. **Notes:** • This bit mask applies to any writes to the addressed byte of any or all of the 4 memory planes, simultaneously. • This bit mask is applicable to any data written into the frame buffer by the CPU, including data that is also subject to rotation, logical functions (AND, OR, XOR), and Set/Reset. To perform a proper read-modify-write cycle into frame buffer, each byte must first be read from the frame buffer by the CPU (and this will cause it to be stored in the memory read latches), this Bit Mask Register must be set, and the new data then written into the frame buffer by the CPU.

GR10 - Address Mapping

I/O (avoid MMIO access) Address:3CFh (Index=10h)

Default:00h

Attributes:R/W

This register should only be accessed using I/O operations and never be accessed through the A/B segment addressing map, I/O space register map, or direct MMIO operations.

Bit	Description
7:4	**Page Select Extension - Unused** These bits form the upper bits of a 12-bit page selection value. When combined with the GR11 <7:0> bits they define the offset into stolen memory to the 64KB page that is accessible via the VGA Memory paging mechanism. These bits are ignored.
3	**Reserved**
2:1	**Paging Map Target.** When paging is enabled, these bits determine the target for data cycle accesses through the VGA memory aperture. VGA graphics memory starts from the base of graphics data stolen memory defined in the PCI configuration BDSM register. VGA display uses the first four 64KB pages of VGA graphics memory. 00 = VGA Graphics Memory 01 = Reserved 10 = Reserved 11 = Reserved

Bit	Description
0	**Page Mapping Enable.** This mode allows the mapping of the VGA memory address space. **Some Notes on Paging.** Once this is enabled, no VGA memory address swizzel will be performed, addresses are directly mapped to memory. A single paging register is used to map the 64KB [A0000:AFFFF] window. An internal address is generated using GR11 as the address lines extension to the lower address lines of the access A[15:2]. When mapping is enabled, the B0000:BFFFF area must be disabled using GR06<3:2>=01. The use of addresses in the A0000-BFFFF range require that both the graphics device PCI configuration memory enable and MSR<1> be enabled. 0 = Disable (default) 1 = Enable

GR11 - Page Selector

I/O Address (avoid MMIO access):3CFh (Index=11h)

Default: 00h

Attributes:R/W

Bit	Description
7	**Reserved**
6:0	**Page Select.** When concatenated with the GR10<7:4> bits, selects a 64KB window within target area when Page Mapping is enabled (GR10[0]=1). This requires that the graphics device PCI configuration space memory enable, the GR06<3:2> bits to be 01 (select A0000-AFFFF only), and the MSR<1:1> bit to be set. This register provides the Address[22:16] bits for the access. VGA paging of frame buffer memory is for non-VGA packed modes only and should not be enabled when using basic VGA modes. This register should only be accessed using I/O operations.

GR18 - Software Flags

I/O (and Memory Offset) Address:3CFh (Index=18h)

Default: 00

Attribute:R/W

Bit	Description
7:0	**Software Flags.** Used as scratch pad space in video BIOS. These bits are separate from the bits which appear in the memory mapped IO space. They are used specifically by the SMI BIOS which does not have access to memory mapped IO at the time they are required. These register bits have no effect on H/W operation.

Attribute Controller Registers

Unlike the other sets of indexed registers, the attribute controller registers are not accessed through a scheme employing entirely separate index and data ports. I/O address 3C0h (or memory address 3C0h) is used both as the read and write for the index register, and as the write address for the data port. I/O address 3C1h (or memory address 3C1h) is the read address for the data port.

To write to the attribute controller registers, the index of the desired register must be written to I/O address 3C0h (or memory address 3C0h), and then the data is written to the very same I/O (memory) address. A flip-flop alternates with each write to I/O address 3C0h (or memory address 3C0h) to change its function from writing the index to writing the actual data, and back again. This flip-flop may be deliberately set so that I/O address 3C0h (or memory address 3C0h) is set to write to the index (which provides a way to set it to a known state) by performing a read operation from Input Status Register 1 (ST01) at I/O address 3BAh (or memory address 3BAh) or 3DAh (or memory address 3DAh), depending on whether the graphics system has been set to emulate an MDA or a CGA as per MSR[0].

To read from the attribute controller registers, the index of the desired register must be written to I/O address 3C0h (or memory address 3C0h), and then the data is read from I/O address 3C1h (or memory address 3C1h). A read operation from I/O address 3C1h (or memory address 3C1h) does not reset the flip-flop to writing to the index. Only a write to 3C0h (or memory address 3C0h) or a read from 3BAh or 3DAh (or memory address 3BAh or 3DAh), as described above, will toggle the flip-flop back to writing to the index.

ARX - Attribute Controller Index Register

I/O (and Memory Offset) Address:3C0h

Default:00UU UUUUb (U=Undefined)

Attributes:Read/Write

Bit	Description
7:6	**Reserved.** Read as 0s.
5	**Video Enable.** Note that In the VGA standard, this is called the "Palette Address Source" bit. Clearing this bit will cause the VGA display data to become all 00 index values. For the default palette, this will cause a black screen. The video timing signals continue. Another control bit will turn video off and stop the data fetches. 0 = Disable. Attribute controller color registers (AR[00:0F]) can be accessed by the CPU. 1 = Enable. Attribute controller color registers (AR[00:0F]) are inaccessible by the CPU.
4:0	**Attribute Controller Register Index.** These five bits are used to select any one of the attribute controller registers (AR[00:14]), to be accessed.

AR[00:0F] - Palette Registers [0:F]

I/O (and Memory Offset) Address:Read at 3C1h and Write at 3C0h; (index=00h-0Fh)

Default:00UU UUUUb (U=Undefined)

Attributes:Read/Write

Bit	Description
7:6	**Reserved.** Read as 0.
5:0	**Palette Bits P[5:0].** In each of these 16 registers, these are the lower 6 of 8 bits that are used to map either text attributes or pixel color input values (for modes that use 16 colors) to the 256 possible colors available to be selected in the palette. **Note:**Bits 3 and 2 of the Color Select Register (AR14) supply bits P7 and P6 for the values contained in all 16 of these registers. Bits 1 and 0 of the Color Select Register (AR14) can also replace bits P5 and P4 for the values contained in all 16 of these registers, if bit 7 of the Mode Control Register (AR10) is set to 1.

AR10 - Mode Control Register

I/O (and Memory Offset) Address:	Read at 3C1h and Write at 3C0h; (index=10h)
Default:	UUh (U=Undefined)
Attributes:	Read/Write

Bit	Description
7	**Palette Bits P5, P4 Select.** **0** = P5 and P4 for each of the 16 selected colors (for modes that use 16 colors) are individually provided by bits 5 and 4 of their corresponding Palette Registers (AR[00:0F]). **1** = P5 and P4 for all 16 of the selected colors (for modes that use 16 colors) are provided by bits 1 and 0 of Color Select Register (AR14).
6	**Pixel Width/Clock Select.** **0** = Six bits of video data (translated from 4 bits via the palette) are output every dot clock. **1** = Two sets of 4 bits of data are assembled to generate 8 bits of video data which is output every other dot clock, and the Palette Registers (AR[00:0F]) are bypassed. **Note:** This bit is set to 0 for all of the standard VGA modes, except mode 13h.
5	**Pixel Panning Compatibility.** **0** = Scroll both the upper and lower screen regions horizontally as specified in the Pixel Panning Register (AR13). **1** = Scroll only the upper screen region horizontally as specified in the Pixel Panning Register (AR13). **Note:** This bit has application only when split-screen mode is being used, where the display area is divided into distinct upper and lower regions which function somewhat like separate displays.
4	**Reserved.** Read as 0.
3	**Enable Blinking/Select Background Intensity.** **0** = Disables blinking in graphics modes, and for text modes, sets bit 7 of the character attribute bytes to control background intensity, instead of blinking. **1** = Enables blinking in graphics modes and for text modes, sets bit 7 of the character attribute bytes to control blinking, instead of background intensity. **Note:** The blinking rate is derived by dividing the VSYNC signal. The Blink Rate Control field of the VGA control register defines the blinking rate.
2	**Enable Line Graphics Character Code.**

Bit	Description
	0 = Every 9th pixel of a horizontal line (i.e., the last pixel of each horizontal line of each 9-pixel wide character box) is assigned the same attributes as the background of the character of which the given pixel is a part. **1** = Every 9th pixel of a horizontal line (i.e., the last pixel of each horizontal line of each 9-pixel wide character box) is assigned the same attributes as the 8th pixel if the character of which the given pixel is a part. This setting is intended to accommodate the line-drawing characters of the PC's extended ASCII character set -- characters with an extended ASCII code in the range of B0h to DFh. **Note:** In some literature describing the VGA standard, the range of extended ASCII codes that are said to include the line-drawing characters is mistakenly specified as C0h to DFh, rather than the correct range of B0h to DFh.
1	**Select Display Type.** **0** = Attribute bytes in text modes are interpreted as they would be for a color display. **1** = Attribute bytes in text modes are interpreted as they would be for a monochrome display.
0	**Graphics/Alphanumeric Mode.** This bit (along with GR06[0]) select either graphics mode or text mode. These two bits must be programmed in a consistent manner to achieve the desired results. **0** = Alphanumeric (text) mode. **1** = Graphics mode.

AR11 - Overscan Color Register

I/O (and Memory Offset) Address:Read at 3C1h and Write at 3C0h; (index=11h)

Default:UUh (U=Undefined)

Attributes:Read/Write

Bit	Description
7:0	**Overscan.** These 8 bits select the overscan (border) color index value. The actual border color will be determined by the contents of the palette at the selected index. The border color is displayed between the end of active and the beginning of blank or the end of blank and the beginning of active on CRT type devices driven from the DAC output port. For native VGA modes on digital display ports there is the option of including the border in the active region or not depending on a control bit in the port control register. For centered VGA modes, the VGA control register determines if the border is included in the centered region or not. For monochrome displays, this value should be set to 00h.

AR12 - Memory Plane Enable Register

I/O (and Memory Offset) Address:Read at 3C1h and Write at 3C0h; (index=12h)

Default:00UU UUUUb (U=Undefined)

Attributes:Read/Write

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7:6</td><td>Reserved. Read as 0.</td></tr>
<tr><td>5:4</td><td>Video Status Mux. These 2 bits are used to select 2 of the 8 possible palette bits (P7-P0) to be made available to be read via bits 5 and 4 of the Input Status Register 1 (ST01). The table below shows the possible choices.

<table>
<tr><th>Bit [5:4]</th><th>ST01 Bit 5</th><th>ST01 Bit 4</th></tr>
<tr><td>00</td><td>P2 (default)</td><td>P0 (default)</td></tr>
<tr><td>01</td><td>P5</td><td>P4</td></tr>
<tr><td>10</td><td>P3</td><td>P1</td></tr>
<tr><td>11</td><td>P7</td><td>P6</td></tr>
</table>
These bits are typically unused by current software; they are provided for EGA compatibility.</td></tr>
<tr><td>3:0</td><td>Enable Plane [3:0]. These 4 bits individually enable the use of each of the 4 memory planes in providing 1 of the 4 bits used in video output to select 1 of 16 possible colors from the palette to be displayed.

0 = Disable the use of the corresponding memory plane in video output to select colors, forcing the bit that the corresponding memory plane would have provided to a value of 0.
1 = Enable the use of the corresponding memory plane in video output to select colors.

Note: AR12 is referred to in the VGA standard as the Color Plane Enable Register.</td></tr>
</table>

AR13 - Horizontal Pixel Panning Register

I/O (and Memory Offset) Address:Read at 3C1h and Write at 3C0h; (index=13h)

Default:0Uh (U=Undefined)

Attributes:Read/Write

<table>
<tr><th>Bit</th><th colspan="5">Description</th></tr>
<tr><td>7:4</td><td colspan="5">Reserved.</td></tr>
<tr><td>3:0</td><td colspan="5">Horizontal Pixel Shift 3-0. This field holds a 4-bit value that selects the number of pixels by which the image is shifted horizontally to the left. This function is available in both text and graphics modes and allows for pixel panning.

In text modes with a 9-pixel wide character box, the image can be shifted up to 9 pixels to the left. In text modes with an 8-pixel wide character box, and in graphics modes other than those with 256 colors, the image can be shifted up to 8 pixels to the left. A pseudo 9-bit mode is when the 9-dot character is selected but overridden by the VGA control bit.

In standard VGA mode 13h (where bit 6 of the Mode Control Register, AR10, is set to 1 to support 256 colors), bit 0 of this register must remain set to 0, and the image may be shifted up to only 4 pixels to the left. In this mode, the number of pixels by which the image is shifted can be further controlled using bits 6 and 5 of the Preset Row Scan Register (CR08).</td></tr>
<tr><td></td><td colspan="5">Number of Pixels Shifted</td></tr>
<tr><td></td><td>Bits [3:0]</td><td>9-dot</td><td>Pseudo 9-dot</td><td>8-dot</td><td>256-Color</td></tr>
<tr><td></td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td></td><td>1</td><td>2</td><td>2</td><td>1</td><td>Undefined</td></tr>
<tr><td></td><td>2</td><td>3</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td></td><td>3</td><td>4</td><td>4</td><td>3</td><td>Undefined</td></tr>
<tr><td></td><td>4</td><td>5</td><td>5</td><td>4</td><td>2</td></tr>
<tr><td></td><td>5</td><td>6</td><td>6</td><td>5</td><td>Undefined</td></tr>
<tr><td></td><td>6</td><td>7</td><td>7</td><td>6</td><td>3</td></tr>
<tr><td></td><td>7</td><td>8</td><td>7</td><td>7</td><td>Undefined</td></tr>
<tr><td></td><td>8</td><td>0</td><td>0</td><td>Undefined</td><td>Undefined</td></tr>
</table>

AR14 - Color Select Register

I/O (and Memory Offset) Address:Read at 3C1h and Write at 3C0h; (index=14h)

Default:0Uh (U=Undefined)

Attributes:Read/Write

Bit	Description
7:4	**Reserved.**
3:2	**Palette Bits P[7:6].** These are the 2 upper-most of the 8 bits that are used to map either text attributes or pixel color input values (for modes that use 16 colors) to the 256 possible colors contained in the palette. These 2 bits are common to all 16 sets of bits P5 through P0 that are individually supplied by Palette Registers 0-F (AR[00:0F]).
1:0	**Alternate Palette Bits P[5:4].** These 2 bits can be used as an alternate version of palette bits P5 and P4. Unlike the P5 and P4 bits that are individually supplied by Palette Registers 0-F (AR[00:0F]), these 2 alternate palette bits are common to all 16 of Palette Registers. Bit 7 of the Mode Control Register (AR10) is used to select between the use of either the P5 and P4 bits that are individually supplied by the 16 Palette Registers or these 2 alternate palette bits.

VGA Color Palette Registers

In devices that have multiple display pipes, there is one palette for each display pipe. These palettes are the same for VGA modes and non-VGA modes. Accesses through VGA register methods can optionally read or write from either one.

For each palette, the color data stored in these 256 color data positions can be accessed only through a complex sub-addressing scheme, using a data register and two index registers. The Palette Data Register at I/O address 3C9h (or memory address offset 3C1h) is the data port. The Palette Read Index Register at I/O address 3C7h (or memory address offset 3C7h) and the Palette Write Index Register at I/O address 3C8h (or memory address offset 3C8h) are the two index registers. The Palette Read Index Register is the index register that is used to choose the color data position that is to be read from via the data port, while the Palette Write Index Register is the index register that is used to choose the color data position that is to be written to through the same data port. This arrangement allows the same data port to be used for reading from and writing to two different color data positions. Reading and writing the color data at a color data position involves three successive reads or writes since the color data stored at each color data position consists of three bytes.

To read a palette color data position, the index of the desired color data position must first be written to the Palette Read Index Register. Then all three bytes of data in a given color data position may be read at the Palette Data Register. The first byte read from the Palette Data Register retrieves the 8-bit value specifying the intensity of the red color component. The second and third bytes read are the corresponding 8-bit values for the green and blue color components respectively. After completing the third read operation, the Palette Read Index Register is automatically incremented so that the data of the next color data position becomes accessible for being read. This allows the contents of all of the 256 color data positions of the palette to be read in sequence. This is done by specifying only the index of the 0th color data position in the Palette Read Index Register, and then simply performing 768 successive reads from the Palette Data Register.

Writing a color data position, entails a very similar procedure. The index of the desired color data position must first be written to the Palette Write Index Register. Then all three bytes of data to specify a given color may be written to the Palette Data Register. The first byte written to the Palette Data Register specifies the intensity of the red color component, the second byte specifies the intensity for the green color component, and the third byte specifies the same for the blue color component. One important detail is that all three of these bytes must be written before the hardware will actually update these three values in the given color data position. When all three bytes have been written, the Palette Write Index Register is automatically incremented so that the data of the next color data position becomes accessible for being written. This allows the contents of all of the 256 color data positions of the palette to be written in sequence. This is done by specifying only the index of the 0th color data position in the Palette Write Index Register, and then simply performing 768 successive writes to the Palette Data Register.

DACMASK - Pixel Data Mask Register

I/O (and Memory Offset) Address:3C6h

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Pixel Data Mask.** In indexed-color mode, the 8 bits of this register are logically ANDed with the 8 bits of pixel data received from the frame buffer for each pixel. The result of this ANDing process becomes the actual index used to select color data positions within the palette. This has the effect of limiting the choice of color data positions that may be specified by the incoming 8-bit data. 0 = Corresponding bit in the resulting 8-bit index being forced to 0. 1 = Allows the corresponding bit in the resulting index to reflect the actual value of the corresponding bit in the incoming 8-bit pixel data.

DACSTATE - DAC State Register

I/O (and Memory Offset) Address:3C7h

Default:00h

Attributes:Read Only

Bit	Description
7:2	**Reserved.** Read as 0.
1:0	**DACState.** This field indicates which of the two index registers was most recently written. **Bits [1:0]** Index Register Indicated 00 = Palette Write Index Register at I/O Address 3C7h (default) 01 = Reserved 10 = Reserved 11 = Palette Read Index Register at I/O Address 3C8h

DACRX - Palette Read Index Register

I/O (and Memory Offset) Address: 3C7h

Default:00h

Attributes:Write Only

Bit	Description
7:0	**Palette Read Index.** The 8-bit index value programmed into this register chooses which of 256 standard color data positions within the palette are to be made accessible for being read from via the Palette Data Register (DACDATA). The index value held in this register is automatically incremented when all three bytes of the color data position selected by the current index have been read. A write to this register will abort a uncompleted palette write sequence. This register allows access to the palette even when running non-VGA display modes.

DACWX - Palette Write Index Register

I/O (and Memory Offset) Address:3C8h

Default:00h

Attributes:Write Only

Bit	Description
7:0	**Palette Write Index.** The 8-bit index value programmed into this register chooses which of 256 standard color data positions within the palette are to be made accessible for being written via the Palette Data Register (DACDATA). The index value held in this register is automatically incremented when all three bytes of the color data position selected by the current index have been written. This register allows access to the palette even when running non-VGA display modes.

DACDATA - Palette Data Register

I/O (and Memory Offset) Address:3C9h

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Palette Data.** This byte-wide data port provides read or write access to the three bytes of data of each color data position selected using the Palette Read Index Register (DACRX) or the Palette Write Index Register (DACWX). The three bytes in each color data position are read or written in three successive read or write operations. The first byte read or written specifies the intensity of the red component of the color specified in the selected color data position. The second byte is for the green component, and the third byte is for the blue component. When writing data to a color data position, all three bytes must be written before the hardware will actually update the three bytes of the selected color data position. When reading or writing to a color data position, ensure that neither the Palette Read Index Register (DACRX) or the Palette Write Index Register (DACWX) are written to before all three bytes are read or written. A write to either of these two registers causes the circuitry that automatically cycles through providing access to the bytes for red, green and blue components to be reset such that the byte for the red component is the one that will be accessed by the next read or write operation via this register. This register allows access to the palette even when running non-VGA display modes. Writes to the palette can cause sparkle if not done during inactive video periods. This sparkle is caused by an attempt to write and read the same address on the same cycle. Anti-sparkle circuits will substitute the previous pixel value for the read output.

CRT Controller Register

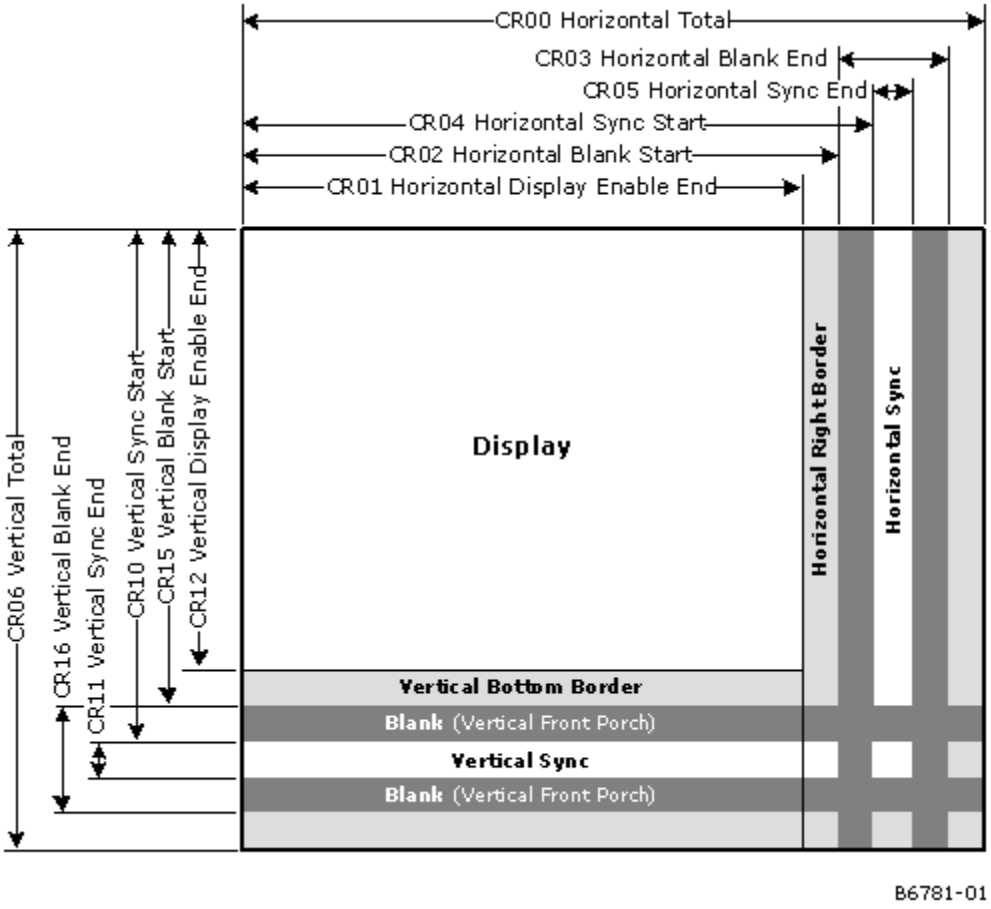
For native VGA modes, the CRTC registers determine the display timing that is to be used. In centered VGA modes, these registers determine the size of the VGA image that is to be centered in the larger timing generator defined rectangle.

The CRT controller registers are accessed by writing the index of the desired register into the CRT Controller Index Register at I/O address 3B4h or 3D4h, depending on whether the graphics system is configured for MDA or CGA emulation. The desired register is then accessed through the data port for the CRT controller registers located at I/O address 3B5h or 3D5h, again depending upon the choice of MDA or CGA emulation as per MSR[0]. For memory mapped accesses, the Index register is at 3B4h (MDA mode) or 3D3h (CGA mode) and the data port is accessed at 3B5h (MDA mode) or 3D5h (CGA mode).

Note:

1. **Group 0 Protection:** In the original VGA, CR[0:7] could be made write-protected by CR11[7]. In BIOS code, this write protection is set following each mode change. Other protection groups have no current use, and would not be used going forward by the BIOS or by drivers. They are the result of an industry fad some years ago to attempt to write protect other groups of registers; however, all such schemes were chip specific. Only the write protection (Group 0 Protection) is supported.

The following figure shows display fields and dimensions and the particular CRxx register that provides the control.


CR00 Horizontal Total
CR03 Horizontal Blank End
CR05 Horizontal Sync End
CR04 Horizontal Sync Start
CR02 Horizontal Blank Start
CR01 Horizontal Display Enable End
CR06 Vertical Total
CR16 Vertical Blank End
CR11 Vertical Sync End
CR10 Vertical Sync Start
CR15 Vertical Blank Start
CR12 Vertical Display Enable End
Display
Horizontal Right Border
Horizontal Sync
Vertical Bottom Border
Blank (Vertical Front Porch)
Vertical Sync
Blank (Vertical Front Porch)
B6781-01

CRX - CRT Controller Index Register

I/O (and Memory Offset) Address:3B4h/3D4h

Default:0Uh (U=Undefined)

Attributes:Read/Write

Bit	Description
7	**Reserved.** Read as 0.
6:0	**CRT Controller Index.** These 7 bits are used to select any one of the CRT controller registers to be accessed via the data port at I/O location 3B5h or 3D5h, depending upon whether the graphics system is configured for MDA or CGA emulation. The data port memory address offsets are 3B5h/3D5h.

CR00 - Horizontal Total Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=00h)

Default:00h

Attributes:Read/Write (Group 0 Protection)

Bit	Description
7:0	**Horizontal Total.** This register is used to specify the total length of each scan line. This encompasses both the part of the scan line that is within the active display area and the part that is outside of it. Programming this register to a zero has the effect of stopping the fetching of display data. This field should be programmed with a value equal to the total number of character clocks within the entire length of a scan line, minus 5.

CR01 - Horizontal Display Enable End Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=01h)

Default:Undefined

Attributes:Read/Write (Group 0 Protection)

Bit	Description
7:0	**Horizontal Display Enable End.** This register is used to specify the end of the part of the scan line that is within the active display area relative to its beginning. In other words, this is the horizontal width of the active display area. This field should be programmed with a value equal to the number of character clocks that occur within the horizontal active display area, minus 1. Horizontal display enable will go active at the beginning of each line during vertical active area, it will go inactive based on the programming of this register or the programming of the horizontal total (CR00) register. When this register value is programmed to a number that is larger than the total number of characters on a line, display enable will be active for all but the last character of the horizontal display line.

CR02 - Horizontal Blanking Start Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=02h)

Default:Undefined

Attributes:Read/Write (Group 0 Protection)

Bit	Description
7:0	**Horizontal Blanking Start.** This register is used to specify the beginning of the horizontal blanking period relative to the beginning of the active display area of a scan line. Horizontal blanking should always be set to start no sooner than after the end of horizontal active. This field should be programmed with a value equal to the number of character clocks that occur on a scan line from the beginning of the active display area to the beginning of the horizontal blanking.

CR03 - Horizontal Blanking End Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=03h)

Default:1UUU UUUUb (U=Undefined)

Attributes:Read/Write (Group 0 Protection)

Bit	Description
7	**Reserved.** Values written to this bit are ignored, and to maintain consistency with the VGA standard, a value of 1 is returned when this bit is read. At one time, this bit was used to enable access to certain light pen registers. At that time, setting this bit to 0 provided this access, but setting this bit to 1 was necessary for normal operation.
6:5	**Display Enable Skew Control.** Defines the degree to which the start and end of the active display area are delayed along the length of a scan line to compensate for internal pipeline delays. These 2 bits describe the delay in terms of a number character clocks. **Bit [6:5]** Amount of Delay 00 = no delay 01 = delayed by 1 character clock 10 = delayed by 2 character clocks 11 = delayed by 3 character clocks
4:0	**Horizontal Blanking End Bits [4:0].** This field provides the 5 least significant bits of a 6-bit value that specifies the end of the blanking period relative to its beginning on a single scan line. Bit 7 of the Horizontal Sync End Register (CR05) supplies the most significant bit. This 6-bit value should be programmed to be equal to the least significant 6 bits of the result of adding the length of the blanking period in terms of character clocks to the value specified in the Horizontal Blanking Start Register (CR02). End of blanking should occur before horizontal total.

CR04 - Horizontal Sync Start Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=04h)

Default:Undefined

Attributes:Read/Write (Group 0 Protection)

Bit	Description
7:0	**Horizontal Sync Start** This register is used to specify the position of the beginning of the horizontal sync pulse relative to the start of the active display area on a scan line. This field should be set equal to the number of character clocks that occur from beginning of the active display area to the beginning of the horizontal sync pulse on a single scan line. Horizontal sync should always occur at least 2 clocks after the start of horizontal blank and 2 clocks before the end of horizontal blank. The actual start of sync will also be affected by both the horizontal sync skew register field and whether it is a text or graphics mode.

CR05 - Horizontal Sync End Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=05h)

Default:00h

Attributes:Read/Write (Group 0 Protection)

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>Horizontal Blanking End Bit 5. This bit provides the most significant bit of a 6-bit value that specifies the end of the horizontal blanking period relative to its beginning. Bits [4:0] of Horizontal Blanking End Register (CR03) supplies the 5 least significant bits. See CR03[4:0] for further details.

This 6-bit value should be set to the least significant 6 bits of the result of adding the length of the blanking period in terms of character clocks to the value specified in the Horizontal Blanking Start Register (CR02).</td></tr>
<tr><td>6:5</td><td>Horizontal Sync Delay. This field defines the degree to which the start and end of the horizontal sync pulse are delayed to compensate for internal pipeline delays. This capability is supplied to implement VGA compatibility. These field describes the delay in terms of a number character clocks.

Bit [6:5] — Amount of Delay
00 — no delay
01 — delayed by 1 character clock
10 — delayed by 2 character clocks
11 — delayed by 3 character clocks</td></tr>
<tr><td>4:0</td><td>Horizontal Sync End. This field provides the 5 least significant bits of a 5-bit value that specifies the end of the horizontal sync pulse relative to its beginning. A value equal to the 5 least significant bits of the horizontal character counter value at which time the horizontal retrace signal becomes inactive (logical 0). Thus, this 5-bit value specifies the width of the horizontal sync pulse. To obtain a retrace signal of W, the following algorithm is used: Value of Horizontal Sync start Register (CR04) + width of horizontal retrace signal in character clock units = 5 bit result to be programmed in this field</td></tr>
</table>

CR06 - Vertical Total Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=06h)

Default:00h

Attributes:Read/Write (Group 0 Protection)

Bit	Description
7:0	**Vertical Total Bits [7:0].** This field provides the 8 least significant bits of either a 10-bit or 12-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area. In standard VGA modes, the vertical total is specified with a 10-bit value. The 8 least significant bits of this value are supplied by these 8 bits of this register, and the 2 most significant bits are supplied by bits 5 and 0 of the Overflow Register (CR07).

CR07 - Overflow Register (Vertical)

I/O (and Memory Offset) Address:3B5h/3D5h (index=07h)

Default:UU0U UUU0b (U=Undefined)

Attributes:Read/Write (Group 0 Protection on bits [7:5, 3:0])

Bit	Description
7	**Vertical Sync Start Bit 9.** The vertical sync start is a 10-bit that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Sync Start Register (CR10), and the most and second-most significant bits are supplied by this bit and bit 2, respectively, of this register. This 10-bit value should be programmed to be equal to the number of scan lines from the beginning of the active display area to the start of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.
6	**Vertical Display Enable End Bit 9.** The vertical display enable end is a 10-bit that specifies the number of the last scan line within the active display area. In standard VGA modes, the vertical display enable end is specified with a 10-bit value. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Display Enable End Register (CR12), and the most and second-most significant bits are supplied by this bit and bit 1, respectively, of this register. This 10-bit value should be programmed to be equal to the number of the last scan line within in the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area, minus 1.
5	**Vertical Total Bit 9.** The vertical total is a 10-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Total Register (CR06), and the most and second-most significant bits are supplied by this bit and bit 0, respectively, of this register. This 10-bit value should be programmed equal to the total number of scan lines, minus 2.

Bit	Description
4	**Line Compare Bit 8.** This bit provides the second most significant bit of a 10-bit value that specifies the scan line at which the memory address counter restarts at the value of 0. Bit 6 of the Maximum Scan Line Register (CR09) supplies the most significant bit, and bits 7-0 of the Line Compare Register (CR18) supply the 8 least significant bits. Normally, this 10-bit value is set to specify a scan line after the last scan line of the active display area. When this 10-bit value is set to specify a scan line within the active display area, it causes that scan line and all subsequent scan lines in the active display area to display video data starting at the very first byte of the frame buffer. The result is what appears to be a screen split into a top and bottom part, with the image in the top part being repeated in the bottom part. When used in cooperation with the Start Address High Register (CR0C) and the Start Address Low Register (CR0D), it is possible to create a split display, as described earlier, but with the top and bottom parts displaying different data. The top part will display what data exists in the frame buffer starting at the address specified in the two aforementioned start address registers, while the bottom part will display what data exists in the frame buffer starting at the first byte of the frame buffer.
3	**Vertical Blanking Start Bit 8.** The vertical blanking start is a 10-bit that specifies the beginning of the vertical blanking period relative to the beginning of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Blanking Start Register (CR15), and the most and second-most significant bits are supplied by bit 5 of the Maximum Scan Line Register (CR09) and this bit of this register, respectively. This 10-bit value should be programmed to be equal to the number of scan lines from the beginning of the active display area to the beginning of the blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical blanking period begins.
2	**Vertical Sync Start Bit 8.** The vertical sync start is a 10-bit value that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Sync Start Register (CR10), and the most and second-most significant bits are supplied by bit 7 and this bit, respectively, of this register. This 10-bit value should be programmed to be equal to the number of scan lines from the beginning of the active display area to the start of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.

Bit	Description
1	**Vertical Display Enable End Bit 8.** The vertical display enable end is a 10-bit value that specifies the number of the last scan line within the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Display Enable End Register (CR12), and the two most significant bits are supplied by bit 6 and this bit, respectively, of this register. This 10-bit or value should be programmed to be equal to the number of the last scan line within in the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area, minus 1.
0	**Vertical Total Bit 8.** The vertical total is a 10-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Total Register (CR06), and the most and second-most significant bits are supplied by bit 5 and this bit, respectively, of this register. This 10-bit value should be programmed to be equal to the total number of scan lines, minus 2.

CR08 - Preset Row Scan Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=08h)

Default:0UUU UUUUb (U=Undefined)

Attributes:Read/Write

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>Reserved. Read as 0s.</td></tr>
<tr><td>6:5</td><td>Byte Panning. This field holds a 2-bit value that selects number of bytes (up to 3) by which the image is shifted horizontally to the left on the screen. This function is available in both text and graphics modes.

In text modes with a 9-pixel wide character box, the image can be shifted up to 27 pixels to the left, in increments of 9 pixels. In text modes with an 8-pixel wide character box, and in all standard VGA graphics modes, the image can be shifted up to 24 pixels to the left, in increments of 8 pixels. When the Nine dot disable bit of the VGA control register is set, the pixel shift will be equivalent to the 8-dot mode.

The image can be shifted still further, in increments of individual pixels, through the use of bits [3:0] of the Horizontal Pixel Panning Register (AR13).
<table>
<tr><th colspan="3">Number of Pixels Shifted</th></tr>
<tr><th>Bit [6:5]</th><th>9-Pixel Text</th><th>8-Pixel Text & Graphics</th></tr>
<tr><td>00</td><td>0</td><td>0</td></tr>
<tr><td>01</td><td>9</td><td>8</td></tr>
<tr><td>10</td><td>18</td><td>16</td></tr>
<tr><td>11</td><td>27</td><td>24</td></tr>
</table>
</td></tr>
<tr><td>4:0</td><td>Starting Row Scan Count. This field specifies which horizontal line of pixels within the character boxes of the characters used on the top-most row of text on the display will be used as the top-most scan line. The horizontal lines of pixels of a character box are numbered from top to bottom, with the top-most line of pixels being number 0. If a horizontal line of the these character boxes other than the top-most line is specified, then the horizontal lines of the character box above the specified line of the character box will not be displayed as part of the top-most row of text characters on the display. Normally, the value specified by these 5 bits should be 0, so that all of the horizontal lines of pixels within these character boxes will be displayed in the top-most row of text, ensuring that the characters in the top-most row of text do not look as though they have been cut off at the top.</td></tr>
</table>

CR09 - Maximum Scan Line Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=09h)

Default:00h

Attributes:Read/Write

Bit	Description
7	**Double Scanning Enable.** 0 = Disable. When disabled, the clock to the row scan counter is equal to the horizontal scan rate. This is the normal setting for many of the standard VGA modes. 1 = Enable. When enabled, the clock to the row scan counter is divided by 2. This is normally used to allow CGA-compatible modes that have only 200 scan lines of active video data to be displayed as 400 scan lines (each scan line is displayed twice).
6	**Line Compare Bit 9.** This bit provides the most significant bit of a 10-bit value that specifies the scan line at which the memory address counter restarts at the value of 0. Bit 4 of the Overflow Register (CR07) supplies the second most significant bit, and bits 7-0 of the Line Compare Register (CR18) supply the 8 least significant bits. Normally, this 10-bit value is set to specify a scan line after the last scan line of the active display area. When this 10-bit value is set to specify a scan line within the active display area, it causes that scan line and all subsequent scan lines in the active display area to display video data starting at the very first byte of the frame buffer. The result is what appears to be a screen split into a top and bottom part, with the image in the top part being repeated in the bottom part. When used in cooperation with the Start Address High Register (CR0C) and the Start Address Low Register (CR0D), it is possible to create a split display, as described earlier, but with the top and bottom parts displaying different data. The top part will display whatever data exists in the frame buffer starting at the address specified in the two aforementioned start address registers, while the bottom part will display whatever data exists in the frame buffer starting at the first byte of the frame buffer.
5	**Vertical Blanking Start Bit 9.** The vertical blanking start is a 10-bit value that specifies the beginning of the vertical blanking period relative to the beginning of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Blanking Start Register (CR15), and the most and second-most significant bits are supplied by this bit and bit 3 of the Overflow Register (CR07), respectively. This 10-bit value should be programmed to be equal to the number of scan line from the beginning of the active display area to the beginning of the blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical blanking period begins.
4:0	**Starting Row Scan Count.** This field provides all 5 bits of a 5-bit value that specifies the number of scan lines in a horizontal row of text. This value should be programmed to be equal to the number of scan lines in a horizontal row of text, minus 1.

CR0A - Text Cursor Start Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=0Ah)

Default:00UU UUUUb (U=Undefined)

Attributes:Read/Write

Bit	Description
7:6	**Reserved.** Read as 0.
5	**Text Cursor Off.** This text cursor exists only in text modes, so this register is entirely ignored in graphics modes. 0 = Enables the text cursor. 1 = Disables the text cursor.
4:0	**Text Cursor Start.** This field specifies which horizontal line of pixels in a character box is to be used to display the first horizontal line of the cursor in text mode. The horizontal lines of pixels in a character box are numbered from top to bottom, with the top-most line being number 0. The value specified by these 5 bits should be the number of the first horizontal line of pixels on which the cursor is to be shown.

CR0B - Text Cursor End Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=0Bh)

Default:0UUU UUUUb (U=Undefined)

Attributes:Read/Write

Bit	Description
7	**Reserved.** Read as 0.
6:5	**Text Cursor Skew.** This field specifies the degree to which the start and end of each horizontal line of pixels making up the cursor is delayed to compensate for internal pipeline delays. These 2 bits describe the delay in terms of a number character clocks. **Bit [6:5]** Amount of Delay 00 = No delay 01 = Delayed by 1 character clock 10 = Delayed by 2 character clocks 11 = Delayed by 3 character clocks
4:0	**Text Cursor End.** This field specifies which horizontal line of pixels in a character box is to be used to display the last horizontal line of the cursor in text mode. The horizontal lines of pixels in a character box are numbered from top to bottom, with the top-most line being number 0. The value specified by these 5 bits should be the number of the last horizontal line of pixels on which the cursor is to be shown.

CR0C - Start Address High Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=0Ch)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Start Address Bits [15:8].**This register provides either bits 15 through 8 of a 16-bit value that specifies the memory address offset from the beginning of the frame buffer at which the data to be shown in the active display area begins. (default is 0) In standard VGA modes, the start address is specified with a 16-bit value. The eight bits of this register provide the eight most significant bits of this value, while the eight bits of the Start Address Low Register (CR0D) provide the eight least significant bits.

CR0D - Start Address Low Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=0Dh)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Start Address Bits [7:0]** This register provides either bits 7 through 0 of a 16 bit value that specifies the memory address offset from the beginning of the frame buffer at which the data to be shown in the active display area begins. (default is 0) In standard VGA modes the start address is specified with a 16-bit value. The eight bits of the Start Address High Register (CR0C) provide the eight most significant bits of this value, while the eight bits of this register provide the eight least significant bits.

CR0E - Text Cursor Location High Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=0Eh)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Text Cursor Location Bits [15:8].** This field provides the 8 most significant bits of a 16-bit value that specifies the address offset from the beginning of the frame buffer at which the text cursor is located. Bit 7:0 of the Text Cursor Location Low Register (CR0F) provide the 8 least significant bits.

CR0F - Text Cursor Location Low Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=0Fh)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Text Cursor Location Bits [7:0].** This field provides the 8 least significant bits of a 16-bit value that specifies the address offset from the beginning of the frame buffer at which the text cursor is located. Bits 7:0 of the Text Cursor Location High Register (CR0E) provide the 8 most significant bits.

CR10 - Vertical Sync Start Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=10h)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Vertical Sync Start Bits [7:0].** This register provides the 8 least significant bits of a 10-bit that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area of a screen. In standard VGA modes, this value is described in 10 bits with bits [7,2] of the Overflow Register (CR07) supplying the 2 most significant bits. This 10-bit value should equal the vertical sync start in terms of the number of scan lines from the beginning of the active display area to the beginning of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.

CR11 - Vertical Sync End Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=11h)

Default:0U00 UUUUb (U=Undefined)

Attributes:Read/Write

Bit	Description
7	**Protect Registers [0:7].** Note that the ability to write to Bit 4 of the Overflow Register (CR07) is not affected by this bit (i.e., bit 4 of the Overflow Register is always writeable). 0 = Enable writes to registers CR[00:07]. (default) 1 = Disable writes to registers CR[00:07].
6	**Reserved.** In the VGA standard, this bit was used to switch between 3 and 5 frame buffer refresh cycles during the time required to draw each horizontal line.
5	**Vertical Interrupt Enable.** This bit is reserved for compatibility only. While this bit may be written or read, it's value will have no effect. Note that the VGA does not provide an interrupt signal which would be connected to an input of the system's interrupt controller. Bit 7 of Input Status Register 0 (ST00) originally indicated the status of the vertical retrace interrupt. 0 = Enable the generation of an interrupt at the beginning of each vertical retrace period. 1 = Disable the generation of an interrupt at the beginning of each vertical retrace period.
4	**Vertical Interrupt Clear.** This is reserved for compatibility only. Note that the VGA does not provide an interrupt signal which would be connected to an input of the system's interrupt controller. 0 = Setting this bit to 0 clears a pending vertical retrace interrupt. This bit must be set back to 1 to enable the generation of another vertical retrace interrupt.
3:0	**Vertical Sync End.** This 4-bit field provides a 4-bit value that specifies the end of the vertical sync pulse relative to its beginning. This 4-bit value should be set to the least significant 4 bits of the result of adding the length of the vertical sync pulse in terms of the number of scan lines that occur within the length of the vertical sync pulse to the value that specifies the beginning of the vertical sync pulse (see the description of the Vertical Sync Start Register for more details).

CR12 - Vertical Display Enable End Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=12h)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Vertical Display Enable End Bits [7:0].** This register provides the 8 least significant bits of a 10-bit value that specifies the number of the last scan line within the active display area. In standard VGA modes, this value is described in 10 bits with bits [6,1] of the Overflow Register (CR07) supplying the two most significant bits. This 10-bit value should be programmed to be equal to the number of the last scan line within in the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area,minus 1.

CR13 - Offset Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=13h)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Offset Bits [7:0].** This register provides either all 8 bits of an 8-bit value that specifies the number of words or DWords of frame buffer memory occupied by each horizontal row of characters. Whether this value is interpreted as the number of words or DWords is determined by the settings of the bits in the Clocking Mode Register (SR01). In standard VGA modes, the offset is described with an 8-bit value, all the bits of which are provided by this register. This 8-bit value should be programmed to be equal to either the number of words or DWords (depending on the setting of the bits in the Clocking Mode Register, SR01) of frame buffer memory that is occupied by each horizontal row of characters.

CR14 - Underline Location Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=14h)

Default:0UUU UUUUb (U=Undefined)

Attributes:Read/Write

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>Reserved. Read as 0.</td></tr>
<tr><td>6</td><td>DWord Mode.
0 = Frame buffer addresses are interpreted by the frame buffer address decoder as being either byte addresses or word addresses, depending on the setting of bit 6 of the CRT Mode Control Register (CR17).
1 = Frame buffer addresses are interpreted by the frame buffer address decoder as being DWord addresses, regardless of the setting of bit 6 of the CRT Mode Control Register (CR17).
Note that this bit is used in conjunction with bits 6 and 5 of the CRT Mode Control Register (CR17) to select how frame buffer addresses from the CPU are interpreted by the frame buffer address decoder as shown below:
<table>
<tr><th>CR14[6]</th><th>CR17[6]</th><th>Addressing Mode</th></tr>
<tr><td>0</td><td>0</td><td>Word Mode</td></tr>
<tr><td>0</td><td>1</td><td>Byte Mode</td></tr>
<tr><td>1</td><td>0</td><td>DWord Mode</td></tr>
<tr><td>1</td><td>1</td><td>DWord Mode</td></tr>
</table></td></tr>
<tr><td>5</td><td>Count By 4.
0 = The memory address counter is incremented either every character clock or every other character clock, depending upon the setting of bit 3 of the CRT Mode Control Register.
1 = The memory address counter is incremented either every 4 character clocks or every 2 character clocks, depending upon the setting of bit 3 of the CRT Mode Control Register. . This is used in mode x13 to allow for using all four planes.
Note that this bit is used in conjunction with bit 3 of the CRT Mode Control Register (CR17) to select the number of character clocks are required to cause the memory address counter to be incremented as shown, below:
<table>
<tr><th>CR14[5]</th><th>CR17[3]</th><th>Addressing Incrementing Interval</th></tr>
<tr><td>0</td><td>0</td><td>every character clock</td></tr>
<tr><td>0</td><td>1</td><td>every 2 character clocks</td></tr>
<tr><td>1</td><td>0</td><td>every 4 character clocks</td></tr>
<tr><td>1</td><td>1</td><td>every 2 character clocks</td></tr>
</table></td></tr>
</table>

Bit	Description
4:0	**Underline Location.** This field specifies which horizontal line of pixels in a character box is to be used to display a character underline in text mode. The horizontal lines of pixels in a character box are numbered from top to bottom, with the top-most line being number 0. The value specified by these 5 bits should be the number of the horizontal line on which the character underline mark is to be shown.

CR15 - Vertical Blanking Start Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=15h)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Vertical Blanking Start Bits [7:0].** This register provides the 8 least significant bits of a 10-bit value that specifies the beginning of the vertical blanking period relative to the beginning of the active display area of the screen. In standard VGA modes, the vertical blanking start is specified with a 10-bit value. The most and second-most significant bits of this value are supplied by bit 5 of the Maximum Scan Line Register (CR09) and bit 3 of the Overflow Register (CR07), respectively. This 10-bit value should be programmed to be equal the number of scan lines from the beginning of the active display area to the beginning of the vertical blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which vertical blanking begins.

CR16 - Vertical Blanking End Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=16h)

Default:Undefined

Attributes:Read/Write

This register provides a 8-bit value that specifies the end of the vertical blanking period relative to its beginning.

Bit	Description
7:0	**Vertical Blanking End Bits [7:0].** This 8-bit value should be set equal to the least significant 8 bits of the result of adding the length of the vertical blanking period in terms of the number of scan lines that occur within the length of the vertical blanking period to the value that specifies the beginning of the vertical blanking period (see the description of the Vertical Blanking Start Register for details).

CR17 - CRT Mode Control

I/O (and Memory Offset) Address:3B5h/3D5h (index=17h)

Default:0UU0 UUUUb (U=Undefined)

Attributes:Read/Write

Bit	Description
7	**CRT Controller Reset.** This bit has no effect except in native VGA modes (non-centered). 0 = Forces horizontal and vertical sync signals to be inactive. No other registers or outputs are affected. 1 = Permits normal operation.
6	**Word Mode or Byte Mode.** 0 = The memory address counter's output bits are shifted by 1 bit position before being passed on to the frame buffer address decoder such that they are made into word-aligned addresses when bit 6 of the Underline Location Register (CR17) is set to 0. 1 = The memory address counter's output bits remain unshifted before being passed on to the frame buffer address decoder such that they remain byte-aligned addresses when bit 6 of the Underline Location Register (CR17) is set to 0. Note that this bit is used in conjunction with bits 6 and 5 of the CRT Mode Control Register (CR17) to control how frame buffer addresses from the memory address counter are interpreted by the frame buffer address decoder as shown below: CR14[6] CR17[6] **Address Mode** 0 0 Word Mode - Addresses from the memory address counter are shifted once to become word-aligned 0 1 Byte Mode - Addresses from the memory address counter are not shifted 1 0 DWord Mode - Addresses from the memory address counter are shifted twice to become DWord-aligned 1 1 DWord Mode - Addresses from the memory address counter are shifted twice to become DWord-aligned
5	**Address Wrap.** Note that this bit is only effective when word mode is made active by setting bit 6 in both the Underline Location Register and this register to 0. 0 = Wrap frame buffer address at 16 KB. This is used in CGA-compatible modes. 1 = No wrapping of frame buffer addresses.

Bit	Description
4	**Reserved.** Read as 0.
3	**Count By 2.** This bit is used in conjunction with bit 5 of the Underline Location Register (CR14) to select the number of character clocks are required to cause the memory address counter to be incremented. 0 = The memory address counter is incremented either every character clock or every 4 character clocks, depending upon the setting of bit 5 of the Underline Location Register. 1 = The memory address counter is incremented either every other clock. CR14[5] CR17[3] Address Incrementing interval 0 0 every character clock 0 1 every 2 character clocks 1 0 every 4 character clocks 1 1 every 2 character clocks
2	**Horizontal Retrace Select.** This bit provides a way of effectively doubling the vertical resolution by allowing the vertical timing counter to be clocked by the horizontal retrace clock divided by 2 (usually, it would be undivided). 0 = The vertical timing counter is clocked by the horizontal retrace clock. 1 = The vertical timing counter is clocked by the horizontal retrace clock divided by 2.
1	**Select Row Scan Counter.** 0 = A substitution takes place, where bit 14 of the 16-bit memory address generated of the memory address counter (after the stage at which these 16 bits may have already been shifted to accommodate word or DWord addressing) is replaced with bit 1 of the row scan counter at a stage just before this address is presented to the frame buffer address decoder. 1 = No substitution takes place. See following tables.
0	**Compatibility Mode Support.** 0 = A substitution takes place, where bit 13 of the 16-bit memory address generated of the memory address counter (after the stage at which these 16 bits may have already been shifted to accommodate word or DWord addressing) is replaced with bit 0 of the row scan counter at a stage just before this address is presented to the frame buffer address decoder. 1 = No substitution takes place. See following tables.

The following tables show the possible ways in which the address bits from the memory address counter can be shifted and/or reorganized before being presented to the frame buffer address decoder. First, the address bits generated by the memory address counter are reorganized, if need be, to accommodate

byte, word or DWord modes. The resulting reorganized outputs (MAOut15-MAOut0) from the memory address counter may also be further manipulated with the substitution of bits from the row scan counter (RSOut1 and RSOut0) before finally being presented to the input bits of the frame buffer address decoder (FBIn15-FBIn0).

Memory Address Counter Address Bits [15:0]

	Byte Mode CR14 bit 6=0 CR17 bit 6=1 CR17 bit 5=X	Word Mode CR14 bit 6=0 CR17 bit 6=0 CR17 bit 5=1	Word Mode CR14 bit 6=0 CR17 bit 6=0 CR17 bit 5=0	DWord Mode CR14 bit 6=1 CR17 bit 6=X CR17 bit 5=X
MAOut0	0	15	13	12
MAOut1	1	0	0	13
MAOut2	2	1	1	0
MAOut3	3	2	2	1
MAOut4	4	3	3	2
MAOut5	5	4	4	3
MAOut6	6	5	5	4
MAOut7	7	6	6	5
MAOut8	8	7	7	6
MAOut9	9	8	8	7
MAOut10	10	9	9	8
MAOut11	11	10	10	9
MAOut12	12	11	11	10
MAOut13	13	12	12	11
MAOut14	14	13	13	12
MAOut15	15	14	14	13

X = Don't Care

Frame Buffer Address Decoder

	CR17 bit 1=1 CR17 bit 0=1	CR17 bit 1=1 CR17 bit 0=0	CR17 bit 1=0 CR17 bit 0=1	CR17 bit 1=0 CR17 bit 0=0
FBIn0	MAOut0	MAOut0	MAOut0	MAOut0
FBIn1	MAOut1	MAOut1	MAOut1	MAOut1
FBIn2	MAOut2	MAOut2	MAOut2	MAOut2
FBIn3	MAOut3	MAOut3	MAOut3	MAOut3
FBIn4	MAOut4	MAOut4	MAOut4	MAOut4
FBIn5	MAOut5	MAOut5	MAOut5	MAOut5
FBIn6	MAOut6	MAOut6	MAOut6	MAOut6

	CR17 bit 1=1 CR17 bit 0=1	CR17 bit 1=1 CR17 bit 0=0	CR17 bit 1=0 CR17 bit 0=1	CR17 bit 1=0 CR17 bit 0=0
FBIn7	MAOut7	MAOut7	MAOut7	MAOut7
FBIn8	MAOut8	MAOut8	MAOut8	MAOut8
FBIn9	MAOut9	MAOut9	MAOut9	MAOut9
FBIn10	MAOut10	MAOut10	MAOut10	MAOut10
FBIn11	MAOut11	MAOut11	MAOut11	MAOut11
FBIn12	MAOut12	MAOut12	MAOut12	MAOut12
FBIn13	MAOut13	MAOut13	RSOut0	RSOut0
FBIn14	MAOut14	RSOut1	MAOut14	RSOut1
FBIn15	MAOut15	MAOut15	MAOut15	MAOut15

CR18 - Line Compare Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=18h)

Default:Undefined

Attributes:Read/Write

Bit	Description
7:0	**Line Compare Bits [7:0].** This register provides the 8 least significant bits of a 10-bit value that specifies the scan line at which the memory address counter restarts at the value of 0. Bit 6 of the Maximum Scan Line Register (CR09) supplies the most significant bit, and bit 4 of the Overflow Register (CR07) supplies the second most significant bit. Normally, this 10-bit value is set to specify a scan line after the last scan line of the active display area. When this 10-bit value is set to specify a scan line within the active display area, it causes that scan line and all subsequent scan lines in the active display area to display video data starting at the very first byte of the frame buffer. The result is what appears to be a screen split into a top and bottom part, with the image in the top part being repeated in the bottom part. (This register is only used in split screening modes, and this is not a problem because split screening is not actually used for extended modes. As a result, there is no benefit to extending the existing overflow bits for higher resolutions.) When used in cooperation with the Start Address High Register (CR0C) and the Start Address Low Register (CR0D), it is possible to create a split display, as described earlier, but with the top and bottom parts displaying different data. The top part will display whatever data exists in the frame buffer starting at the address specified in the two aforementioned start address registers, while the bottom part will display whatever data exists in the frame buffer starting at the first byte of the frame buffer.

CR22 - Memory Read Latch Data Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=22h)

Default:00h

Attributes:Read Only

Bit	Description
7:0	**Memory Read Latch Data.** This field provides the value currently stored in 1 of the four memory read latches. Bits 1 and 0 of the Read Map Select Register (GR04) select which of the four memory read latches may be read via this register.

CR24 - Toggle State of Attribute Controller Register

I/O (and Memory Offset) Address:3B5h/3D5h (index=24h)

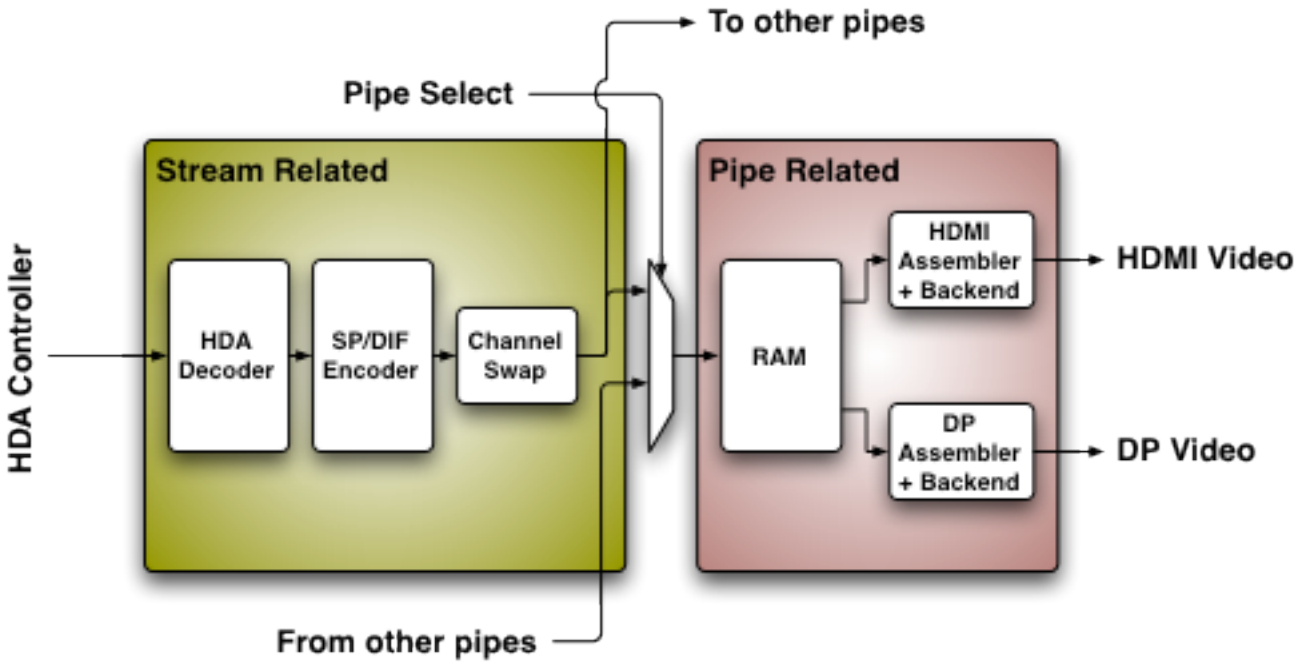
Default:00h

Attributes:Read Only

Bit	Description
7	**Toggle Status.** Indicates where the last write to attribute register was to: 0 = index port 1 = data port
6:0	**Reserved.** Read as 0.

Display Audio Codec Verbs

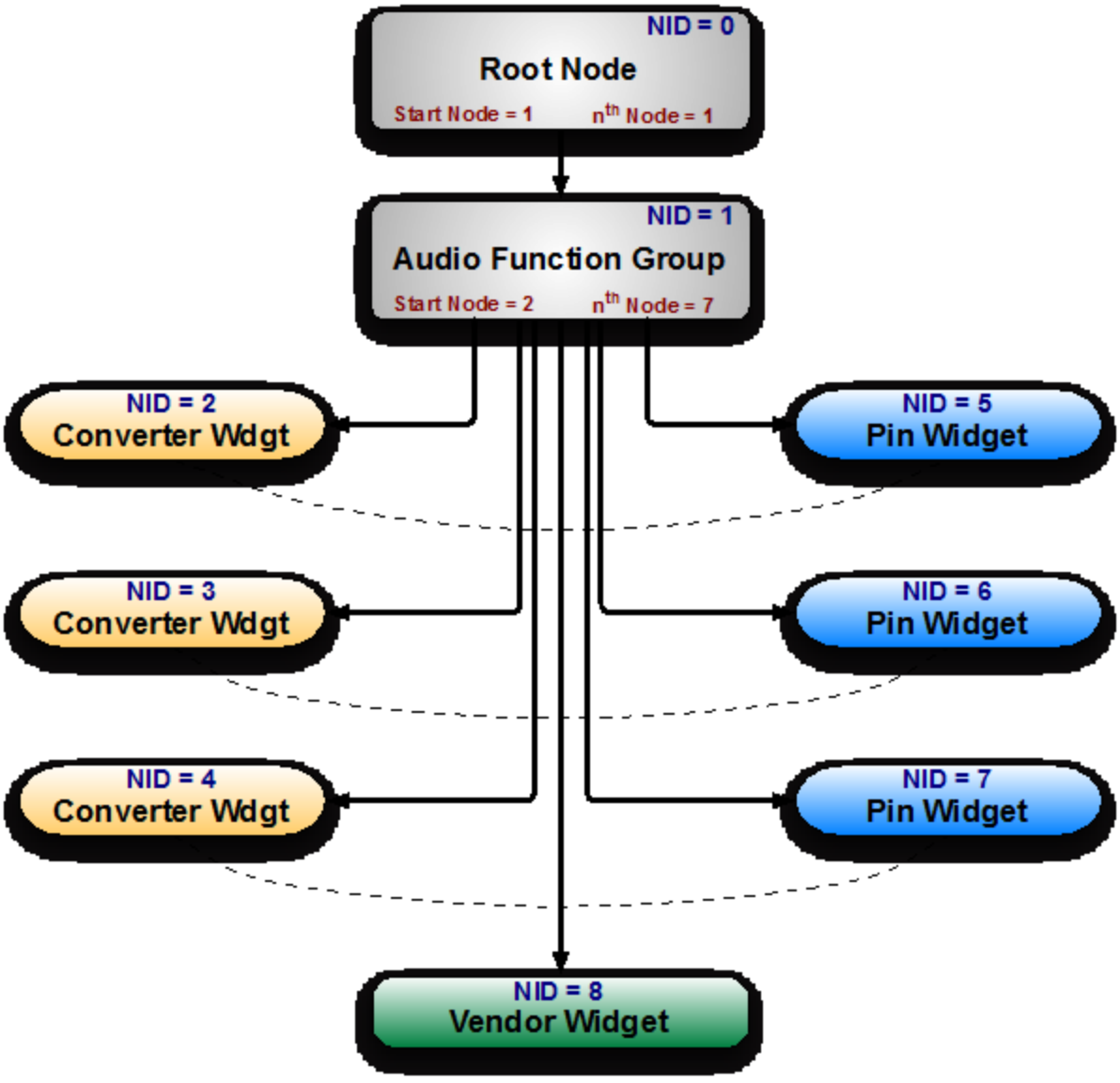
Block Diagram



Codec Node Hierarchy

The diagram below shows the hierarchy of the internal codec. The codec is presented as a single codec with multiple endpoints. By operating as a single codec, only one driver needs to be loaded on the system.

Inside the codec are three "converter widgets" and three "pin widgets", responsible for taking data from HD Audio DMA engines and placing into an HDMI/DP stream. Each pin widget has a 1-1 connection to a converter widget (as indicated by the dotted lines in the diagram).



Programming

Programming of the codec is performed by "verbs" as described in the HD Audio specification. These verbs travel over the internal HD Audio link at a rate of 1 verb per frame. A verb can either come from the CORB, with responses using the RIRB, or using an immediate command and response mechanism (ICR). Device 2 contains its own copy of an ICR mechanism as a back-door into the audio codec.

Verb Support

Verb ID		Verb Name	Node ID							
Set	Get		01h	02h	03h	04h	05h	06h	07h	08h
2h	Ah	Stream Descriptor Format		Y	Y	Y				
3h	Bh	Set Amplifier Mute					Y	Y	Y	
-	F00h	Get Parameters	Y	Y	Y	Y	Y	Y	Y	Y
701h	F01h	Connection Select Control					Y	Y	Y	
-	F02h	Connection List Entry					Y	Y	Y	
705h	F05h	Power State		Y	Y	Y	Y	Y	Y	
706h	F06h	Channel and Stream ID		Y	Y	Y				
707h	F07h	Pin Widget Control					Y	Y	Y	
708h	F08h	Unsolicited Response Enable					Y	Y	Y	
-	F09h	Pin Sense					Y	Y	Y	
-	F0Dh	Digital Converter		Y	Y	Y				
70Dh	-	Digital Converter 1		Y	Y	Y				
70Eh	-	Digital Converter 2		Y	Y	Y				
-	F1Ch	Configuration Default					Y	Y	Y	
71Ch	-	Configuration Default Byte 0					Y	Y	Y	
71Dh	-	Configuration Default Byte 1					Y	Y	Y	
71Eh	-	Configuration Default Byte 2					Y	Y	Y	
71Fh	-	Configuration Default Byte 3					Y	Y	Y	
-	F20h	Subsystem ID	Y							
-	F21h	Subsystem ID	Y							
-	F22h	Subsystem ID	Y							
-	F23h	Subsystem ID	Y							
720h	-	Subsystem ID[7: 0]	Y							
721h	-	Subsystem ID[15: 8]	Y							
722h	-	Subsystem ID[23:16]	Y							
723h	-	Subsystem ID[31:24]	Y							
72Dh	F2Dh	Converter Channel Count		Y	Y	Y				
-	F2Eh	HDMI/DP Info Size					Y	Y	Y	
730h	F30h	HDMI Info Index					Y	Y	Y	
731h	F31h	HDMI Info Data					Y	Y	Y	
732h	F32h	HDMI Info Transmit Control					Y	Y	Y	
733h	F33h	Protection Control					Y	Y	Y	
734h	F34h	Converter Channel Map					Y	Y	Y	

Verb ID		Verb Name	Node ID							
Set	Get		01h	02h	03h	04h	05h	06h	07h	08h
735h	F35h	Device Select					Y	Y	Y	
-	F36h	Display Device List Entry					Y	Y	Y	
73Ch	73Ch	DisplayPort Stream ID					Y	Y	Y	
73Eh	-	Digital Converter 3		Y	Y	Y				
73Fh	-	Digital Converter 4		Y	Y	Y				
-	F80h	HDMI / DP Status								Y
781h	F81h	HDMI Vendor Verb								Y
782h	-	GTC Capture Trigger								Y
-	F83h	Captured Wall Clock Value								Y
-	F84h	Captured GTC Value								Y
-	F85h	Get GTC Offset Value								Y
785h	-	Set GTC Offset Value[7: 0]								Y
786h	-	Set GTC Offset Value[15: 8]								Y
787h	-	Set GTC Offset Value[23:16]								Y
788h	-	Set GTC Offset Value[31:24]								Y
789h	F89h	Converter Channel Count								Y

Parameter Support

Param ID	Parameter Name	Node ID								
		00h	01h	02h	03h	04h	05h	06h	07h	08h
00h	Vendor ID	Y								
02h	Revision ID	Y								
04h	Subordinate Node Count	Y	Y							
05h	Function Group Type		Y							
08h	Audio Function Group Capabilities									
09h	Audio Widget Capabilities			Y	Y	Y	Y	Y	Y	Y
0Ah	Sample Size, Rate CAPs			Y	Y	Y				
0Bh	Stream Formats			Y	Y	Y				
0Ch	Pin Capabilities						Y	Y	Y	
0Dh	Input Amp Capabilities									
0Eh	Connection List Length						Y	Y	Y	
0Fh	Supported Power States		Y							
10h	Processing Capabilities									
11h	GPIO Count									
12h	Output Amp Capabilities						Y	Y	Y	
13h	Volume Knob Capabilities									
15h	Device List Length						Y	Y	Y	

Node ID 00h: Root Node Verbs

The root node only contains a single verb - the "Get Parameters" verb at F00h.

F00h: Get Parameters

Parameter	Symbol	Register Name
04h	PARAM_SNC	Subordinate Node Count
05h	PARAM_FGT	Function Group Type
08h	PARAM SPS	Function Group Capabilities
0Fh	PARAM SPS	Supported Power States

Parameter 04h: PARAM_SNC - Subordinate Node Count

Bit	Reset	Description
31:24	0	*Reserved*
23:16	02h	**Start Node Number (SNN):** Indicates the start node number of widget or functional nodes in the Functional Group.
15:08	0	*Reserved*
07:00	07h	**Total Number of Nodes (TNN):** Indicates 7 widgets in the Functional Group. (HDMI/DP converters (3) + HDMI/DP pins (3) + Vendor Defined Widget (1))

Parameter 05h: PARAM_FGT - Function Group Type

Bit	Reset	Description
31:09	0	*Reserved*
08	0	**Unsolicited Capable (UC):** Not capable of generating an unsolicited response.
07:00	01h	**Node Type (NT):** Indicates Audio Function Group.

Parameter 08h: PARAM_FGC - Function Group Capability

Bit	Reset	Description
31:04	0	*Reserved*
03:00	00h	**Output Delay (OD)**Output Delay.

Parameter 0Fh: PARAM_SPS - Supported Power States

Bit	Reset	Description
31	1	**Extended Power State Supported (EPSS):** Indicates support for low power states
30:04	0	*Reserved*
03	1	**D3 Supported (D3S):** Indicates support for D3.
02	0	**D2 Supported (D2S):** Indicates no support for D2.
01	0	**D1 Supported (D1S):** Indicates no support for D1.
00	1	**D0 Supported (D0S):** Indicates support for D0.

Node ID 01h: Audio Function Group Verbs

Set Verb	Get Verb	Symbol	Name
-	F00h	GET PARAM	Get Parameters
705h	F05h	SET_PS / GET_PS	Set Power State
-	F20h	GET_SSID	Get Subsystem ID
720h	720h	SET_SSID0	Set Subsystem ID
721h	721h	SET_SSID1	Set Subsystem ID
722h	722h	SET_SSID2	Set Subsystem ID
723h	723h	SET_SSID3	Set Subsystem ID

F00h: Get Parameters

Parameter	Symbol	Register Name
04h	PARAM_SNC	Subordinate Node Count
05h	PARAM_FGT	Function Group Type
08h	PARAM SPS	Function Group Capabilities
0Fh	PARAM SPS	Supported Power States

Parameter 04h: PARAM_SNC - Subordinate Node Count

Bit	Reset	Description
31:24	0	*Reserved*
23:16	02h	**Start Node Number (SNN):** Indicates the start node number of widget or functional nodes in the Functional Group.
15:08	0	*Reserved*
07:00	07h	**Total Number of Nodes (TNN):** Indicates 7 widgets in the Functional Group. (HDMI/DP converters (3) + HDMI/DP pins (3) + Vendor Defined Widget (1))

Parameter 05h: PARAM_FGT - Function Group Type

Bit	Reset	Description
31:09	0	*Reserved*
08	0	**Unsolicited Capable (UC):** Not capable of generating an unsolicited response.
07:00	01h	**Node Type (NT):** Indicates Audio Function Group.

Parameter 08h: PARAM_FGC - Function Group Capability

Bit	Reset	Description
31:04	0	*Reserved*
03:00	00h	**Output Delay (OD)**Output Delay.

Parameter 0Fh: PARAM_SPS - Supported Power States

Bit	Reset	Description
31	1	**Extended Power State Supported (EPSS):** Indicates support for low power states
30:04	0	*Reserved*
03	1	**D3 Supported (D3S):** Indicates support for D3.
02	0	**D2 Supported (D2S):** Indicates no support for D2.
01	0	**D1 Supported (D1S):** Indicates no support for D1.
00	1	**D0 Supported (D0S):** Indicates support for D0.

705h: SET_PS - Set Power State

Bits	Description
07:02	Reserved
01:00	**Requested Power State (RPS):** Only D0 (00) and D3 (11) may be requested

F05h: GET_PS - Get Power State

Bits	Reset	Description
31:11	0	Reserved
10	0	**Settings Reset (SR):** Haswell does not change the default values.
09	1	**Clock Stop OK (CSOK):** Clock stopping in D3 is OK
08	0	**Error (ERR):** No error will ever be reported.
07:06	0	Reserved
05:04	11	**Actual Power State (APS):** Indicates the current power state of the node.
03:02	0	Reserved
01:00	11	**Requested Power State (CPS):** Reflects value written with SET_PS verb.

F20h: GET SSID - Get Subsystem ID0

Bits	Reset	Description
31:00	80860101h	**Subsystem ID (SSID):** Reports the sub-system ID set via SET_SSIDx verbs.

F21h: GET SSID - Get Subsystem ID1

Bits	Reset	Description
31:00	80860101h	**Subsystem ID (SSID):** Reports the sub-system ID set via SET_SSIDx verbs.

F22h: GET SSID - Get Subsystem ID2

Bits	Reset	Description
31:00	80860101h	**Subsystem ID (SSID):** Reports the sub-system ID set via SET_SSIDx verbs.

F23h: GET SSID - Get Subsystem ID3

Bits	Reset	Description
31:00	80860101h	**Subsystem ID (SSID):** Reports the sub-system ID set via SET_SSIDx verbs.

720h: SET SSID0 - Set Subsystem ID0

Bits	Description
07:00	Subsystem ID Bits [7:0]

721h: SET SSID1 - Set Subsystem ID1

Bits	Description
07:00	Subsystem ID Bits [15:8]

722h: SET SSID2 - Set Subsystem ID2

Bits	Description
07:00	Subsystem ID Bits [23:16]

723h: SET SSID3 - Set Subsystem ID3

Bits	Description
07:00	Subsystem ID Bits [31:24]

Node ID 02h, 03h, 04h: Audio Output Convertor Widget Verbs

Verb	Symbol	Verb Name
2h	SET_SDF	Set Stream Descriptor Format
Ah	GET_SDF	Get Stream Descriptor Format
F00h	GET_PARAM	Get Parameters
705h	SET_PS	Set Power State
F05h	GET_PS	Get Power State
706h	SET_CSID	Set Channel and Stream ID
F06h	GET_CSID	Get Channel and Stream ID
F0Dh	SET_DC1	Get Digital Converter
70Dh	SET_DC1	Set Digital Converter 1
70Eh	SET_DC2	Set Digital Converter 2
73Eh	SET_DC3	Set Digital Converter 3
73Fh	SET_DC4	Set Digital Converter 4
72Dh	SET_CCC	Set Converter Channel Count
F2Dh	GET_CCC	Get Converter Channel Count

2h/Ah: SET/GET_SDF - Set/GET Stream Descriptor Format

Bits	Reset	Description
31:15	0	Reserved
14	0	**Sample Base Rate (SBR):** Hardwired to 0. (48 kHz)
13:11	000	**Sample Base Rate Multiplier (SBRM):** Hardwired to 000 (x1)
10:08	000	**Sample Base Rate Divisor (SBRD):** Hardwired to 000 (divide by 1)
07	0	Reserved
06:04	011	**Bits / Sample (BPS):** • 001b: Data is packed in memory in 16 bit containers on 16 bit boundaries • 010b: Data is packed in memory in 20 bit containers on 32 bit boundaries • 011b: Data is packed in memory in 24 bit containers on 24 bit boundaries • 100b: Data is packed in memory in 32 bit containers on 32 bit boundaries All other bit combinations reserved
03:00	1h	**# Channels in Stream (NCS):** 2 channels in each frame

F00h: Get Parameters

Parameter	Symbol	Register Name
09h	PARAM_AWC	Audio Widget Capabilities
0Ah	PARAM_PSB	Parameter Sizes and Bit Rates
0Bh	PARAM_SF	Stream Formats
0Fh	PARAM_SPS	Supported Power States

Parameter 09h: AWC - Audio Widget Capabilities

Bits	Reset	Description
31:24	0	Reserved
23:20	0h	**Widget Type (TYPE):** Indicates this is an audio output widget
19:16		**Sample Delay in Widget (DELAY):**
15:13	011	**Channel Count Extension (CCE):** These three bits, combined with STRO, indicate that there are 8 channels supported.
11	0	**L-R Swap (LRS):** Indicates no left/right channel swap.
10	1	**Power Control (PC):** Indicates power state control
09	1	**Digital (DIG):** Indicates support for digital streams.
08	0	**Connection List (CL):** Indicates no connection list
07	0	**Unsolicited Capable (UC):** Indicates support for unsolicited responses.
06	0	**Processing Widget (PW):** Indicates no support for processing
05	0	**Stripe (STRP):** Indicates striping not supported.
04	1	**Format Override (FO):** Indicates support for formatting
03	0	**Amp Parameter Override (APO):** Indicates no amplifier support.
02	0	**Out Amp Present (OAP):** Indicates no output amplifier present.
01	0	**In Amp Present (IAP):** Indicates no input amplifier present.
00	1	**Stereo (STRO):** Indicates a stereo widget

Parameter 0Ah: PSB - PCM Sizes and Bit Rates

Bits	Reset	Description
31:21	0	*Reserved*
20	1	**32-bit Support (B32):** Indicates 32-bit samples supported
19	1	**24-bit Support (B24):** Indicates 24-bit samples supported
18	1	**20-bit Support (B20):** Indicates 20-bit samples supported
17	1	**16-bit Support (B16):** Indicates 16-bit samples supported
16	0	**8-bit Support (B8):** Indicates 8-bit samples not supported
15:12	0	*Reserved*
11	0	**384 kHz Support (R12):** Indicates 384 kHz not supported
10	1	**192 kHz Support (R11):** Indicates 192 kHz supported
09	1	**176.4 kHz Support (R10):** Indicates 176.4 kHz supported
08	1	**96 kHz Support (R9):** Indicates 96 kHz supported
07	1	**88.2 kHz Support (R8):** Indicates 88.2 kHz supported
06	1	**48 kHz Support (R7):** Indicates 48 kHz supported
05	1	**44.1 kHz Support (R6):** Indicates 44.1 kHz supported
04	1	**32 kHz Support (R5):** Indicates 32 kHz supported
03	0	**22.05 kHz Support (R4):** Indicates 22.05 kHz not supported
02	0	**16 kHz Support (R3):** Indicates 16 kHz not supported
01	0	**11.025 kHz Support (R2):** Indicates 11.025 kHz not supported
00	0	**8 kHz Support (R1):** Indicates 8 kHz not supported

Parameter 0Bh: SF - Stream Formats

Bits	Reset	Description
31:03	0	*Reserved*
02	1	**AC3 Support (AC3):** Indicates AC3 stream format is supported
01	0	**Float32 Support (F32):** Indicates float32 stream format not supported
00	1	**PCM Support (PCM):** Indicates PCM format is supported.

Parameter 0Fh: PARAM_SPS - Supported Power States

Bit	Reset	Description
31	1	**Extended Power State Supported (EPSS):** Indicates support for low power states
30:04	0	*Reserved*
03	1	**D3 Supported (D3S):** Indicates support for D3.
02	0	**D2 Supported (D2S):** Indicates no support for D2.
01	0	**D1 Supported (D1S):** Indicates no support for D1.
00	1	**D0 Supported (D0S):** Indicates support for D0.

705h: SET_PS - Set Power State

Bits	Description
07:02	Reserved
01:00	**Requested Power State (RPS):** Only D0 (00) and D3 (11) may be requested

F05h: GET_PS - Get Power State

Bits	Reset	Description
31:11	0	Reserved
10	0	**Settings Reset (SR):**
09	0	**Clock Stop OK (CSOK):** Clock stopping in D3 is not OK

Bits	Reset	Description
08	0	**Error (ERR):** No error will ever be reported.
07:06	0	Reserved
05:04	11	**Actual Power State (APS):** Indicates the current power state of the node.
03:02	0	Reserved
01:00	11	**Requested Power State (CPS):** Reflects value written with SET_PS verb.

706h/F06h: GET/SET_CSID - Get/Set Channel & Stream ID

Bits	Reset	Description
07:04	0h	**Stream ID (SID):** Link stream used by the converter for data output.
03:00	0h	**Lowest Channel Number (LCN):** Lowest channel used by the converter.

Digital Converter Verbs

F0Dh: GET_DC - Get Digital Converter

Bits	Reset	Description
31:24	0	Reserved
23	1	**Keep Alive (KA):** See SET_DC3.KA
22:20	0	Reserved
19:16	0h	**IEC Coding Type (ICT):** See SET_DC3.ICT
15	0	Reserved
14:08	00h	**Category Code (CC):** See SET_DC1.CC
07	0	**Level (LVL):** See SET_DC1.LVL
06	0	**Professional (PRO):** See SET_DC1.PRO
05	0	**Audio is not PCM (AUDIO):** See SET_DC1.AUDIO
04	0	**Copyright (COPY):** See SET_DC1.COPY
03	0	**Pre-emphasis (PRE):** See SET_DC1.PRE
02	0	**Validity Configuration (VCFG):** See SET_DC1.VCFG
01	0	**Validity (V):** See SET_DC1.V
00	1	**Digital Enable (DIGEN):** See SET_DC1.DIGEN

70Dh: SET_DC1 - Set Digital Converter 1

Bits	Description
07	**Level (LVL):** S/PDIF IEC Generation Level.
06	**Professional (PRO):** When set, indicates professional use of channel.
05	**Audio is not PCM (AUDIO):** When set, data is non-PCM format.
04	**Copyright (COPY):** A 1 indicates content may be copied freely, 0 indicates copyright asserted.
03	**Pre-emphasis (PRE):** When set, enables filter pre-emphasis.
02	**Validity Configuration (VCFG):** Determines S/PDIF transmitter behavior when data is not being transmitted.
01	**Validity (V):** Affects the validity flag transmitted in each sub-frame, and enables S/PDIF transmitter to maintain connection during error or mute conditions.
00	**Digital Enable (DIGEN):** When set, enables digital content

70Eh: Digital Converter 2

Bits	Description
07	Reserved
06:00	**Category Code (CC):** S/PDIF IEC Category Code.

73Eh: Digital Converter 3

Bits	Description
07	Keep Alive
06:04	Reserved
03:00	IEC Coding Type

73Fh: Digital Converter 4

Bits	Description
07:00	Reserved

72Dh/F2Dh: GET/SET_CCC - Get/Set Converter Channel Count

Bits	Reset	Description
07:04	0	Reserved
03:00		Converter Channel Count 1 (0^{th} order)

Node ID 05h, 06h, 07h: Pin Widget Verbs

Set Verb	Get Verb	Symbol	Verb Name
3h	-	SET_AM	Set Amplifier Mute
-	Bh	GET_AM	Get Amplifier Mute
-	F00h	-	Get Parameters
701h	F01h	SET_CSC / GET_CSC	Set/Get Connection Select Control
-	F02h	-	Get Connection List Entry
705h	F05h	SET_PS / GET_PS	Set/Get Power State
707h	F07h	SET_PWC / GET_PWC	Set/Get Pin Widget Control
708h	F08h	SET_UE / GET_UE	Set/Get Unsolicited Response Enable
-	F09h	-	Get Pin Sense
71Ch	-	SET_CD0	Set Configuration Default Byte 0
71Dh	-	SET_CD1	Set Configuration Default Byte 1
71Eh	-	SET_CD2	Set Configuration Default Byte 2
71Fh	-	SET_CD3	Set Configuration Default Byte 3
-	F1Ch	GET_CD	Get Configuration Default
-	F2Eh	GET_HDIS	Get HDMI/DP Info Size
730h	F30h	SET_HII / GET_HII	Set/Get HDMI Info Index
731h	F31h	SET_HID / GET_HID	Set/Get HDMI Info Data
732h	F32h	SET_HITC / GET_HITC	Set/Get HDMI Info Transmit Control
733h	F33h	SET_PC / GET_PC	Set/Get Protection Control
734h	F34h	SET_CCM / GET_CCM	Set/Get Converter Channel Map
735h	F35h	SET_DS / GET_DS	Set/Get Device Select
-	F36h	GET_DDLE	Get Display Device List Entry
73Ch	F3Ch	SET_DPID / GET_DPID	Set/Get DisplayPort Stream ID

3h: SET_AM - Set Amplifier Mute

Bits	Bits	Description
15	0	**Set Output Amp (SOA):**.
14	0	**Set Input Amp (SIA):**.
13	0	**Set Left Amp (SLA):**.
12	0	**Set Right Amp (SRA):**.
11:08	0h	**Index (IDX):**
07	0	**Mute (MUTE):** When set, amp muted.
06:00	0	*Reserved*

B8h: GET_AM - Get Amplifier Mute

Bits	Bits	Description
31:08	0	*Reserved*
07	0	**Mute (MUTE):** When set, amp muted.
06:00	0	*Reserved*

F00h: Get Parameters

Parameter	Symbol	Register Name
09h	PARAM_AWC	Audio Widget Capabilities
0Ch	PARAM_PC	Pin Capabilities
0Eh	PARAM_CLL	Connection List Length
12h	PARAM OAC	Output Amplifier Capabilities
15h	PARAM_DLL	Device List Length
0Fh	PARAM_SPS	Supported Power States

Parameter 09h: AWC - Audio Widget Capabilities

Bits	Reset	Description
31:24	0	Reserved
23:20	4h	**Widget Type (TYPE):** Indicates this is a pin complex widget
19:16	0	**Sample Delay in Widget (DELAY):** No delay through the pin widget.
15:13	011	**Channel Count Extension (CCE):** This field, combined with STRO, indicate 8 channels supported.
11	0	**L-R Swap (LRS):** Indicates no left/right channel swap.
10	1	**Power Control (PC):** Indicates power state control
09	1	**Digital (DIG):** Indicates support for digital streams.
08	1	**Connection List (CL):** Indicates a connection list
07	1	**Unsolicited Capable (UC):** Indicates support for unsolicited responses.
06	0	**Processing Widget (PW):** Indicates no support for processing
05	0	**Stripe (STRP):** Indicates striping not supported.
04	0	**Format Override (FO):** Indicates no support for formatting
03	0	**Amp Parameter Override (APO):** Indicates no amplifier override support.
02	0	**Out Amp Present (OAP):** Indicates no output amplifier present.
01	0	**In Amp Present (IAP):** Indicates no input amplifier present.
00	1	**Stereo (STRO):** Indicates a stereo widget

Parameter 0Ch: PC - Pin Capabilities

Bits	Reset	Description
31:28	0	*Reserved*
27	1	**High Bit Rate (HBR):** Indicates support for high bit-rate audio
26:25	0	*Reserved*
24	1	**DisplayPort (DP):** Indicates support for DisplayPort
23:08	0	*Reserved*
07	1	**HDMI (HDMI):** Indicates support for HDMI
06:05	0	*Reserved*
04	1	**Output Capable (OC):** Pin is output capable
03	0	*Reserved*
02	1	**Presence Detect Capable (PDC):** Indicates capability for presence detection
01:00	0	*Reserved*

Parameter 0Eh: CLL - Connection List Length

Bits	Reset	Description
31:08	0	*Reserved*
07	0	**Long Form (LF):** Indicates connection list is short form
06:00	01h	**Length (LEN):** Indicates there is one item in the connection list.

Parameter 12h: OAC - Output Amplifier Capabilities

Bits	Reset	Description
31	1	**Mute Capable (MC):** Muting is capable on this pin
30:00	0	Reserved

Parameter 15h: DLL - Device List Length

Bits	Reset	Description
31:06	0	*Reserved*
05:00	00h	**Length (LEN):** Indicates no devices

Parameter 0Fh: PARAM_SPS - Supported Power States

Bit	Reset	Description
31	1	**Extended Power State Supported (EPSS):** Indicates support for low power states
30:04	0	*Reserved*
03	1	**D3 Supported (D3S):** Indicates support for D3.
02	0	**D2 Supported (D2S):** Indicates no support for D2.
01	0	**D1 Supported (D1S):** Indicates no support for D1.
00	1	**D0 Supported (D0S):** Indicates support for D0.

701h/F01h: SET/GET_CSC - Set/Get Connection Select Control

Bits	Reset	Description
07:00	00h	**Connection Select Control (CSC):**

F02h: GET_CLE - Get Connection List Entry

Bits	Reset	Description
31:08	0	Reserved
07:00	Varies	**Connection List Entry (CLE):** 02h for NodeID 05h, 03h for NodeID 06h, and 04h for NodeID 07h.

705h: SET_PS - Set Power State

Bits	Description
07:02	Reserved
01:00	**Requested Power State (RPS):** Only D0 (00) and D3 (11) may be requested

F05h: GET_PS - Get Power State

Bits	Reset	Description
31:11	0	Reserved
10	0	**Settings Reset (SR):** Haswell does not change the default values.
09	0	**Clock Stop OK (CSOK):** Clock stopping in D3 is not OK
08	0	**Error (ERR):** No error will ever be reported.
07:06	0	Reserved
05:04	11	**Actual Power State (APS):** Indicates the current power state of the node.
03:02	0	Reserved
01:00	11	**Requested Power State (CPS):** Reflects value written with SET_PS verb.

707h/F07h: SET/GET_PWC - Set/Get Pin Widget Control

Bits	Reset	Description
07	0	Reserved
06	0	**Out Enable (OE):** When set, the audio is enabled
05:02	0	Reserved
01:00	00	**Encoded Packet Type (EPT):**

708h/F08h: SET/GET_UE - Set/Get Unsolicited Enable

Bits	Description
07	**Unsolicited Enable (UE):** When set, unsolicited responses are allowed
06	Reserved
05:00	**Tag (TAG):**

F09h: GET_PS - Get Pin Sense

Bits	Reset	Description
31	0	**Presence Detect (PD):** When set presence is detected on this pin.
30	0	**ELD Value (ELDV):**
29	0	**Inactive (INA):**
28:00	28:00	Reserved

71Ch: SET_CD0 - Set Configuration Default Byte 0

Bits	Description
07:04	Default Association (DA):
03:00	Sequence (SEQ):

71Dh: SET_CD1 - Set Configuration Default Byte 1

Bits	Description
07:04	Color (COL):
03:00	Miscellaneous (MISC):

71Eh: SET_CD2 - Set Configuration Default Byte 2

Bits	Description
07:04	Default Device (DD):
03:00	Connection Type (CT):

71Fh: SET_CD3 - Set Configuration Default Byte 3

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>07:06</td><td>Port Connectivity (PC): External connectivity of the pin complex.
00 = Connected to jack
01 = No physical connection
10 = Fixed function device (integrated speaker, mic, etc.)
11 = Both a jack and internal connection</td></tr>
<tr><td>05:00</td><td>Location (LOC):
<table>
<tr><td></td><td colspan="10">Bits 3:0</td></tr>
<tr><td>Bits 5:4</td><td>0h: N/A</td><td>1h: Rear</td><td>2h: Front</td><td>3h: Left</td><td>4h: Right</td><td>5h: Top</td><td>6h: Bottom</td><td>7h: Special</td><td>8h: Special</td><td>9h: Special</td><td>Ah-Fh Reserved</td></tr>
<tr><td>00: External</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td></td><td></td></tr>
<tr><td>01: Internal</td><td>Y</td><td></td><td></td><td></td><td></td><td></td><td></td><td>Y</td><td>Y</td><td>Y</td><td></td></tr>
<tr><td>10: Separate Chassis</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td></td><td></td><td></td><td></td></tr>
<tr><td>11: Other</td><td>Y</td><td></td><td></td><td></td><td></td><td></td><td>Y</td><td>Y</td><td>Y</td><td></td><td></td></tr>
</table></td></tr>
</table>

F1Ch: GET_CD - Get Configuration Default

Bits	Description
31:30	**Port Connectivity (PC):** See SET_CD3.PC
29:24	**Location (L):** See SET_CD3.L
23:20	**Default Device (DD):** See Set_CD2.DD
19:16	**Connection Type (CT):** See Set_CD2.CT
15:12	**Color (COL):** See SET_CD1.COL
11:08	**Miscellaneous (MISC):** See SET_CD1.MISC
07:04	**Default Association (DA):** See SET_CD0.DA
03:00	**Sequence (SEQ):** See SET_CD0.SEQ

F2Eh: HDMI/DP Info Size

Bits	Reset	Description
31:08	0	Reset
07:00	Varies	**Size (SZ):** Indexes 0 - 3 return 1Eh, index 1000 returns 54h, others reserved.

730h/F30h: SET/GET_HII - Set/Get HDMI Info Index

Bits	Reset	Description
07:05	000	**Infoframe Packet Index (IPI):** Value / Name / Value / Name 000 / Audio / 011 / GP3 001 / GP / 100 / GP4 010 / GP2 / Others / Reserved
04:00	00h	**Byte Offset Index Pointer (BOI):**

731h/F31h: SET/GET_HID - Set/Get HDMI Info Data

Bits	Reset	Description
07:00	00h	**Data (DATA):** Data at current index pointed to from SET_HII verb.

732h/F32h: SET/GET_HITC - Set/Get HDMI Info Transmit Control

Bits	Reset	Description
07:06	00	**InfoFrame Control Current Indexed Frame (IFCCIF):** • 00 = Disable Transmit • 01 = Reserved • 10 = Transmit Once • 11 = Best Effort
05:00	0	*Reserved*

734h/F34h: SET/GET_CCM - Get/Set Converter Channel Map

Bits	Reset	Description
07:04	0h	Converter Channel (CC):
03:00	0h	Slot (SN):

735h: SET_DS - Set Device Select

Bits	Reset	Description
07:06	0	*Reserved*
05:00	00h	**Device (D):** 000001, 000010 (based upon number of devices present)

F35h: GET_DS - Get Device Select

Bits	Description
31:12	**Reserved:** Set to 0
11:06	**SinK Device ID:** Sink Device ID in the multi stream topology of the DP hierarchy.
05:00	**Device (D):**Device Entry index currently set

>

F36h: GET_DDLE - Get Display Device List Entry

Bits	Bits	Description
31:08	0	*Reserved*
07	0	**Reserved**
06	0	**IA of Entry 1**
05	0	**ELDV of Entry 1**
04	0	**PD of Entry 1**
03	0	**Reserved**
02	0	**IA of Entry 0**
01	0	**ELDV of Entry 0**
00	0	**PD of Entry 0**

73Ch/F3Ch: SET/GET_DPID - Set/Get DisplayPort Stream ID

Bits	Reset	Description
07:03	00h	**Tag (TAG):** Represents the SSID that will go in the lower 5 bits of the SSID
02:00	000	**Index (IDX):** Pointer to program multiple SSID

Display High Definition Audio Controller

All audio controller registers (including the memory mapped registers) must be addressable as byte, word, and D-word quantities. Software must always make register accesses on natural boundaries; D-word accesses must be on D-word boundaries, word accesses on word boundaries, etc.

Note that the Intel HD Audio memory-mapped register space must not be accessed with the LOCK semantic exclusive-access mechanism. If software attempts exclusive-access mechanisms to the Intel HD Audio memory-mapped register space, the results are undefined.

All registers not mentioned are reserved. Reserved registers will always read 00h and writes will have no effect. Software must properly handle reserved bits. Unless specified, reserved bits must be preserved using read-modify-writes; whilst reserved bits specified as "RsvdZ" bits must be written as zeros. This behavior helps to ensure future compatibility.

PCI Configuration Space

The Intel Display HD Audio controller is a PCI Express device. It is Device #3, Function #0. If the Intel Display HD Audio controller is disabled via fuses or a Function Disable bit, configuration accesses to the Intel Display HD Audio register space are ignored as if the device is not present.

Unless specified, all the registers in this section are reset by platform reset, D3HOT to D0 reset, and FLR.

PCI Header

VID_DID - Vendor Defined ID and Device ID

CMD_STS - Command and Status

CLASS - Revision ID, Programming Interface, Sub Class Code and Base Class Code

CLS - Cache Line Size, Latency Timer, Header Type and Built in Self Test

DHDALBAR - Display HD Audio Lower Base Address

DHDAUBAR - Display HD Audio Upper Base Address

SVID_SID - Subsystem Vendor ID and SubSystem ID

CAPPTR - Capabilities Pointer

INTLN_INTPN - Interrupt Line and Interrupt Pin

PCI Power Management Capability Structure

PID_PC - Power Management Capability ID and Capabilities

PCS - Power Management Control and Status

MSI Capability Structure

MID_MMC - MSI Cap ID and Message Control

MMA - MSI Message Base Address

MMD - MSI Message Data

PCI Express Capability Structure

PXID_PXC - PCI Express Cap ID and Control

DEVCAP - Device Capabilities

DEVC_DEVS - Device Control and Status

MMIO Registers

These are Display Audio Controller MMIO registers as defined in the HD Audio Spec.

GCAP_VMIN_VMAJ - Global Capabilities, Minor and Major Version

OUTPAY_INPAY - Output Payload and Input payload Capability

GCTL - Global Control

WAKEEN_WAKESTS - Wake Enable and Wake Status

GSTS - Global Status

OUTSTRMPAY_INSTRMPAY - Output/Input Stream Payload Capability

INTCTL - Interrupt Control

INTSTS - Interrupt Status

WALCLK - Wall Clock Counter

SSYNC - Stream Synchronization

CORBLBASE - CORB (Command Output Ring Buffer) - Lower Base Address

CORBUBASE - CORB (Command Output Ring Buffer) - Upper Base Address

CORBRWP - CORB Read/Write Pointers

CORBCTL_STS_SIZE - CORB Control_Status_Size

RIRBLBASE - RIRB (Response Input Ring Buffer) - Lower Base Address

RIRBUBASE - RIRB (Response Input Ring Buffer) - Upper Base Address

RIRBWP_RINTCNT - RIRB Write Pointer and Interrupt Count

RIRBCTL_STS_SIZE - RIRB Control, Status and Size

ICOI - Immediate Command Output Interface

IRII - Immediate Response Input Interface

ICS - Immediate Command Status

DPLBASE - DMA Position Lower Base Address

DPUBASE - DMA Position Upper Base Address

SDCTL_STS - Output Stream Descriptor Control and Status

SDLPIB - Output Stream Descriptor Link Position in Current Buffer

SDCBL - Output Stream Descriptor Cyclic Buffer Length

SDLVI - Output Stream Descriptor Last Valid Index

SDFIFOD_FMT - Output Stream Descriptor FIFO Data and Format

SDBDPL - Output Stream Descriptor Buffer Descriptor List Pointer Lower

SDBDPU - Output Stream Descriptor Buffer Descriptor List Pointer Upper

EM4 - Extended Mode 4

EM5 - Extended Mode 5

DPIB - DMA Position in Buffer

WALCLKA - Wall Clock Counter Alias

SDLPIBA - Output Stream Descriptor Link Position in Current Buffer Alias

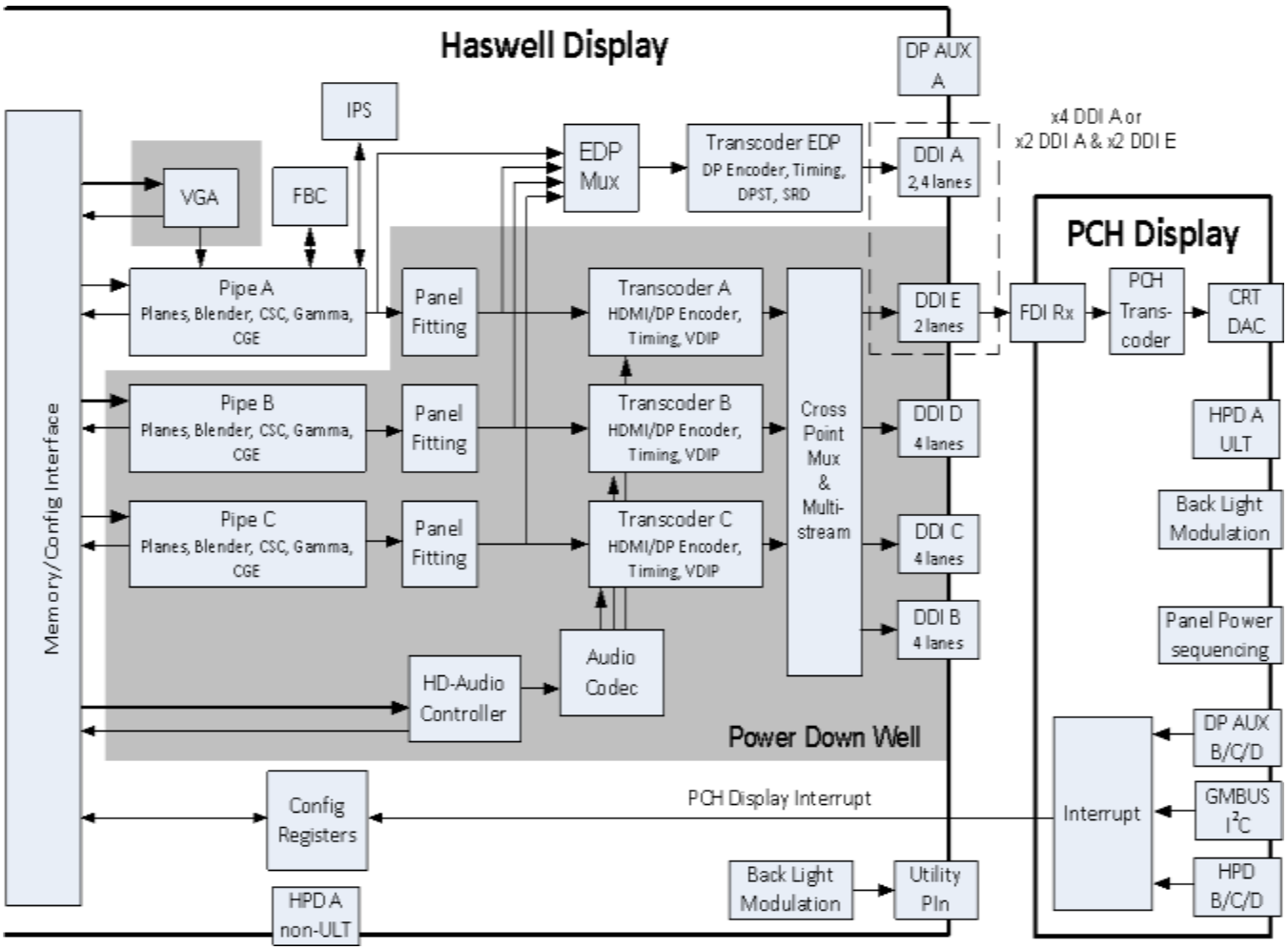
North Display Engine Registers

This chapter contains the register descriptions for the display portion of a family of graphics devices.

These registers vary by devices within the family of devices, so special attention needs to be paid to which devices use which registers and register fields.

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

Haswell Display Connections



The front end of the display contains the pipes. There are three instances which are referred to as Pipe A, Pipe B, and Pipe C.

Intermediate Pixel Storage (IPS) is only supported on Haswell ULT.

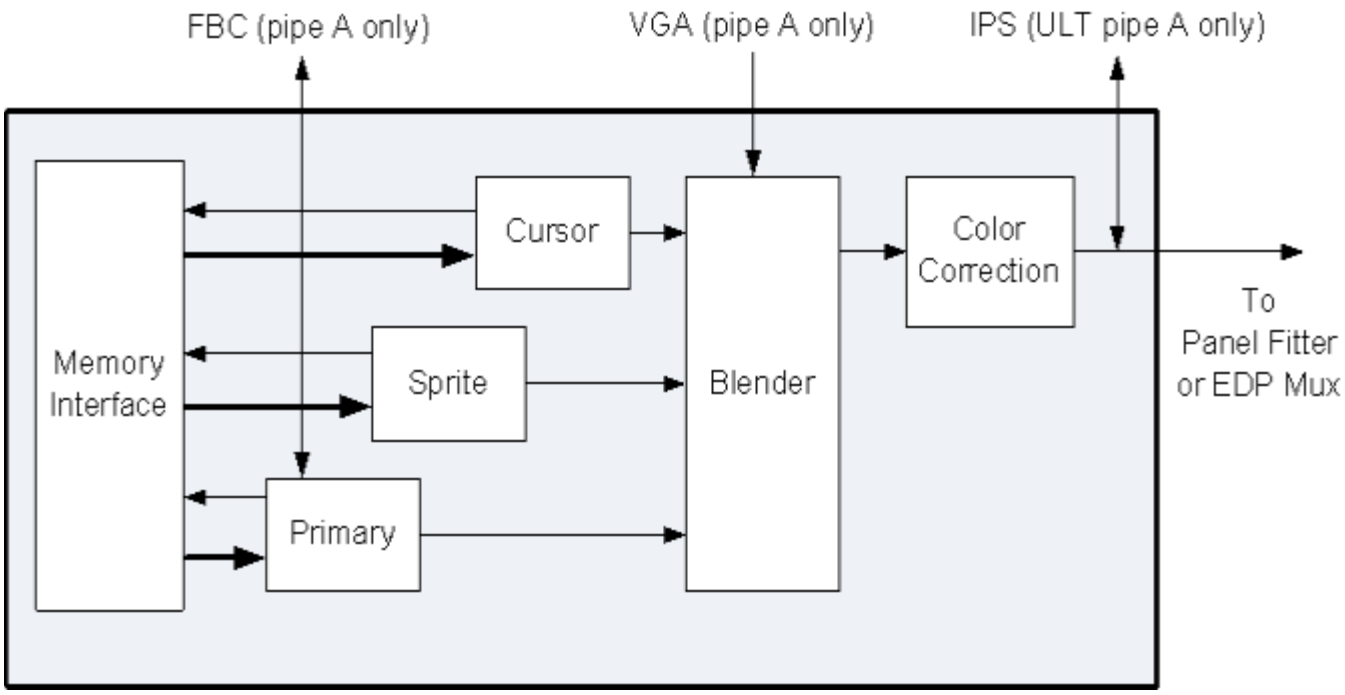
The pipes connect to the panel fitters. The pipe A panel fitter may be bypassed when going to DDI A.

The panel fitters connect to the transcoders. There are four instances which are referred to as Transcoder A, Transcoder B, Transcoder C, and Transcoder EDP. They are also sometimes called Pipe A, B, C, and EDP when referring to the timing generator logic or DDI Slice A, B, C, and EDP when referring to the port logic.

The transcoders connect to the DDIs. There are five instances which are referred to as DDI A, DDI B, DDI C, DDI D, and DDI E.

DDI D is not supported on Haswell ULT.

Display Pipes



The display pipes contain the planes, blending, pipe CSC, pipe gamma, and pipe color gamut enhancement.

Each display pipe has a primary plane, a sprite plane, and a cursor.

The planes are blended in a fixed Z-order with the primary plane on the bottom, sprite plane in the middle, and cursor on the top.

Display Transcoders

The display transcoders contain the timing generators, dithering and clipping, HDMI/DVI/DisplayPort encoders, Audio/Video mixers, and Video Data Island Packet mixers.

Transcoder EDP supports does not support HDMI, DVI, or Audio. Only Transcoder EDP supports Panel Self Refresh and DPST.

Audio

The audio controller streams audio data from memory to the audio codec.

The audio codec connects to the transcoders.

DDIs

The DDIs contain the DisplayPort transport control and other port logic to interface to the DDI physical pins.

DDI A, DDI B, DDI C, and DDI D support lane reversal where the internal lane to package lane mapping is swapped.

DDI A and DDI E do not support DisplayPort multistream.

DDI D is not supported on Haswell ULT.

DDI A and DDI E share 2 lanes. DDI A is capable of supporting up to 4 lanes when DDI E is not connected, but only 2 lanes when DDI E is connected. DDI E is capable of supporting up to 2 lanes when connected. Dynamic switching between the two configurations is not supported.

DDI B, DDI C, and DDI D Lane Mapping:

Lanes on Package	Package Lane 0	Package Lane 1	Package Lane 2	Package Lane 3
Internal Lanes without Reversal	DDI 0	DDI 1	DDI 2	DDI 3
Internal Lanes with Reversal	DDI 3	DDI 2	DDI 1	DDI 0

DDI A and DDI E Lane Mapping:

Lanes on Package	Package Lane 0	Package Lane 1	Package Lane 2	Package Lane 3
Internal Lanes without Reversal when DDI A is x4 capable	DDI A 0	DDI A 1	DDI A 2	DDI A 3
Internal Lanes with Reversal when DDI A is x4 capable	DDI A 3	DDI A 2	DDI A 1	DDI A 0
Internal Lanes without Reversal when DDI A is x2 capable	DDI A 0	DDI A 1	DDI E 0	DDI E 1
Internal Lanes with Reversal when DDI A is x2 capable	DDI A 1	DDI A 0	DDI E 0	DDI E 1

DDI Equivalent Names:

DDI Name	Equivalent Names
DDI A	Port 0, DDI-A, DDIA, Port A, DDI0, eDP
DDI B	Port 1, DDI-B, DDIB, Port B, DDI1
DDI C	Port 2, DDI-C, DDIC, Port C, DDI2
DDI D Only on HSW non-ULT	Port 3, DDI-D, DDID, Port D, DDI3
DDI E	Port 4, DDI-E, DDIE, Port E, DDI4

FDI

The FDI mode of operation allows a DDI to connect to the PCH display to send output to the CRT DAC.

Only one DDI can operate in FDI mode at a time.

DDI E is preferred for FDI use. DDI A is not recommended for FDI use.

FDI is only supported with 1 or 2 lanes enabled and a 2.7 GHz bit clock using the PCH SSC reference.

Pipe to Transcoder to DDI Mappings

Twin modes are not supported.

Any pipe can drive any single DDI.

With DisplayPort multistream it is possible to have multiple pipes driving a single DDI. DDI B, DDI C, and DDI D support multistream. DDI A and DDI E do not support multistream.

Pipe A can connect to either Transcoder A or Transcoder EDP, but not both simultaneously.

Pipe A can take two paths to reach Transcoder EDP. The path that bypasses panel fitting, avoiding the power down well, is used when the power down well will be turned off for increased power savings.

Pipe B can connect to either Transcoder B or Transcoder EDP, but not both simultaneously.

Pipe C can connect to either Transcoder C or Transcoder EDP, but not both simultaneously.

Transcoder A is tied to Pipe A.

Transcoder B is tied to Pipe B.

Transcoder C is tied to Pipe C.

Transcoder EDP can connect to Pipe A, Pipe B, or Pipe C, but only one at a time.

Transcoder A can connect to DDI B, DDI C, DDI D, or DDI E, but only one at a time.

Transcoder B can connect to DDI B, DDI C, DDI D, or DDI E, but only one at a time.

Transcoder C can connect to DDI B, DDI C, DDI D, or DDI E, but only one at a time.

Transcoder EDP can connect only to DDI A.

DDI A can connect only to Transcoder EDP. DDI A does not support DisplayPort multistream.

DDI B can connect to Transcoder A, Transcoder B, or Transcoder C, individually or simultaneously if DisplayPort multistream is used.

DDI C can connect to Transcoder A, Transcoder B, or Transcoder C, individually or simultaneously if DisplayPort multistream is used.

DDI D can connect to Transcoder A, Transcoder B, or Transcoder C, individually or simultaneously if DisplayPort multistream is used.

DDI E can connect to Transcoder A, Transcoder B, or Transcoder C, individually.

Display Resolution Support

A display resolution is only supported if it meets all the restrictions below for Maximum Pipe Pixel Rate, Maximum Port Link Rate, Maximum Port Pixel Rate, Maximum Memory Read Bandwidth, and Maximum Watermark.

Core Display Clock (CDCLK)

Attribute	Haswell	Haswell ULT	Haswell ULX
CDCLK	540 MHz	450 MHz	337.5 MHz

Maximum Pipe Pixel Rate

The display resolution must fit within the maximum pixel rate output from the pipe.

For each plane (primary and sprite) {

    Plane Ratio = 1

    If plane is enabled and source pixel format is 64 bits per pixel {

        If sprite and primary planes are both enabled on the same pipe {Plane Ratio = 8/10}

        Else {Plane Ratio = 8/9}

    }

}

Pipe Ratio = Minimum[Sprite Plane Ratio, Primary Plane Ratio]

If panel fitting is enabled {

    Horizontal down scale amount = Maximum[1, panel fitter window horizontal size / pipe horizontal source size]

    Vertical down scale amount = Maximum[1, panel fitter window vertical size / pipe vertical source size]

    Down scale amount = Horizontal down scale amount * Vertical down scale amount

    Pipe Ratio = Pipe Ratio * Down scale amount

}

Pipe maximum pixel rate = CDCLK frequency * Pipe Ratio

Maximum Port Link Rate

The display resolution must fit within the maximum link rate for each port type and processor.

Attribute	Haswell	Haswell ULT	Haswell ULX
eDP/DP	HBR2 540 MHz	HBR2 540 MHz	HBR 270 MHz
HDMI	300 MHz	300 MHz	300 MHz
DVI	165 MHz	165 MHz	165 MHz
FDI	270 MHz	270 MHz	270 MHz
CRT DAC	180 MHz	180 MHz	180 MHz

Maximum Port Pixel Rate

The display resolution must fit within the maximum pixel rate for each port type and processor.

Attribute	Haswell	Haswell ULT	Haswell ULX
eDP/DP x4 single stream	540 Mpps 18, 24, 30bpp (CDCLK limited) 480 Mpps 36bpp	450 Mpps 18, 24, 30, 36bpp (CDCLK limited)	337.5 Mpps 18, 24bpp (CDCLK limited) 288 Mpps 30bpp 240 Mpps 36bpp
eDP/DP x2 single stream	480 Mpps 18bpp 360 Mpps 24bpp 288 Mpps 30bpp 240 Mpps 36bpp	450 Mpps 18bpp (CDCLK limited) 360 Mpps 24bpp 288 Mpps 30bpp 240 Mpps 36bpp	240 Mpps 18bpp 180 Mpps 24bpp 144 Mpps 30bpp 120 Mpps 36bpp
eDP/DP x1 single stream	240 Mpps 18bpp 180 Mpps 24bpp 144 Mpps 30bpp 120 Mpps 36bpp	240 Mpps 18bpp 180 Mpps 24bpp 144 Mpps 30bpp 120 Mpps 36bpp	120 Mpps 18bpp 90 Mpps 24bpp 72 Mpps 30bpp 60 Mpps 36bpp
DP multistream	<= 540 Mpps per stream while considering total link bandwidth	<= 450 Mpps per stream while considering total link bandwidth	<= 337.5 Mpps per stream while considering total link bandwidth
HDMI	300 Mpps 24bpp 200 Mpps 36bpp	300 Mpps 24bpp 200 Mpps 36bpp	300 Mpps 24bpp 200 Mpps 36bpp
DVI	165 Mpps 24bpp	165 Mpps 24bpp	165 Mpps 24bpp
CRT DAC	180 Mpps 24bpp	180 Mpps 24bpp	180 Mpps 24bpp

Maximum Memory Read Bandwidth

The display resolution must not exceed the available system memory bandwidth, considering factors like thermal throttling and bandwidth available for other memory clients.

For each pipe {

    For each plane (primary and sprite) { // cursor can be ignored

        Plane bandwidth MB/s = pixel rate MHz * source pixel format in bytes * Down scale amount

        Total display bandwidth MB/s = Total display bandwidth + Plane bandwidth

    }

}

If Total display bandwidth > system memory bandwidth available for display {Bandwidth exceeded = 1}

Maximum Watermark

The display resolution must not exceed the WM_PIPE maximum watermark value. See the volume on Watermark Programming.

Display Resolution Capabilities

These resolutions meet all the resolution restrictions for up to 3 simultaneous displays, 4 primary or sprite planes with 32bpp pixel format, and 1 cursor, with no panel fitter down scaling.

Attribute	Haswell	Haswell ULT	Haswell ULX
eDP/DP x4 single stream	3840x2160 60Hz 30bpp[4] 4096x2160 30Hz 30bpp[2]	3200x2000 60Hz 30bpp[3] 4096x2160 30Hz 30bpp[2] 3840x2160 30Hz 30bpp[1]	2880x1620 60Hz 24bpp[2] 4096x2160 30Hz 30bpp[2] 2560x1600 60Hz 30bpp[1] 3840x2160 30Hz 30bpp[1]
HDMI	4096x2304 24Hz 24bpp[2] 3840x2160 30Hz 24bpp[1]	4096x2304 24Hz 24bpp[2] 3840x2160 30Hz 24bpp[1]	4096x2304 24Hz 24bpp[2] 3840x2160 30Hz 24bpp[1]
DVI	1920x1200 60Hz 24bpp[1]	1920x1200 60Hz 24bpp[1]	1920x1200 60Hz 24bpp[1]
CRT DAC	1920x1200 60Hz 24bpp[1]	1920x1200 60Hz 24bpp[1]	1920x1200 60Hz 24bpp[1]

[1]Requires at least single channel DDR3 1333 for 3 simultaneous displays

[2]Requires at least single channel DDR3 1600 for 3 simultaneous displays

[3]Requires at least dual channel DDR3 1333 for 3 simultaneous displays

[4]Requires at least dual channel DDR3 1600 for 3 simultaneous displays

Examples

Example pipe pixel rate:

Primary plane enabled at 32bpp, sprite plane enabled at 16bpp, panel fitting enabled and down scale amount 0.89, and CDCLK 450 MHz:

Primary ratio = 1

Sprite ratio = 1

Pipe ratio = Minimum[1, 1] = 1

Pipe ratio = 1 * 0.89 = 0.89

Pipe maximum pixel rate = 450 MHz * 0.89 = 400.5 MHz

Example pipe pixel rate:

Primary plane enabled at 64bpp, sprite plane enabled at 32bpp, no panel fitting enabled, and CDCLK 540 MHz:

Primary ratio = 8/10

Sprite ratio = 1

Pipe ratio = Minimum[1, 8/10] = 8/10

Pipe maximum pixel rate = 540 MHz * 8/10 = 432 MHz

Example memory bandwidth:

System memory bandwidth available for display = 4000 MB/s

Pipe A - Primary plane enabled at 32bpp, sprite plane enabled at 16bpp, panel fitting disabled, pixel rate 148.5 MHz

Pipe B - Primary plane enabled at 32bpp, panel fitting disabled, pixel rate 148.5 MHz

Pipe C - Primary plane enabled at 32bpp, panel fitting disabled, pixel rate 148.5 MHz

Pipe A - Primary bandwidth = 148.5 * 4 bytes = 594 MB/s

Pipe A - Sprite bandwidth = 148.5 * 2 bytes = 297 MB/s

Pipe B - Primary bandwidth = 148.5 * 4 bytes = 594 MB/s

Pipe C - Primary bandwidth = 148.5 * 4 bytes = 594 MB/s

Total display bandwidth = 594 + 297 + 594 + 594 = 2079 MB/s

System memory bandwidth available for display not exceeded

Example memory bandwidth:

System memory bandwidth available for display = 4000 MB/s

Pipe A - Primary plane enabled at 32bpp, sprite plane enabled at 32bpp, panel fitting enabled and down scale amount 0.89, pixel rate 414.5 MHz

Pipe B - Primary plane enabled at 32bpp, panel fitting disabled, pixel rate 414.5 MHz

Pipe C - Primary plane enabled at 32bpp, panel fitting disabled, pixel rate 414.5 MHz

Pipe A - Primary bandwidth = 414.5 * 4 bytes * 1/0.89 = 1863 MB/s

Pipe A - Sprite bandwidth = 414.5 * 4 bytes * 1/0.89 = 1863 MB/s

Pipe B - Primary bandwidth = 414.5 * 4 bytes = 1658 MB/s

Pipe C - Primary bandwidth = 414.5 * 4 bytes = 1658 MB/s

Total display bandwidth = 1863 + 1863 + 1658 + 1658 = 7042 MB/s

System memory bandwidth available for display **exceeded**

Terminology

Term	Description
DP	DisplayPort
SST, DP SST	DisplayPort Single Stream Transport
MST, DP MST	DisplayPort Multi Stream Transport

Register Access Field	Description	Implementation
R/W (Read/Write)	The value written into this register will control hardware and is the same value that will be read.	Write data is stored. Read is from the stored data. Stored value is used to control hardware.
Reserved	Unused register bit. Don't assume a value for these bits. Writes have no effect.	Write data is ignored. Read is zero.
MBZ (Must Be Zero)	Always write a zero to this register.	May be implemented as Reserved or as R/W.
PBC (Preserve Bit Contents)	Software must write the original value back to this bit. This allows new features to be added using these bits.	May be implemented as Reserved or as R/W.
Read Only	The read value is determined by hardware. Writes to this bit have no effect.	Write data is ignored. Read is from a status signal or some other internal source.
Write Only	The value written into this register will control hardware. Reads return zero.	Write data is stored. Read is zero. Stored value is used to control hardware.
R/W Clear (Read/Write Clear)	Sticky status bit. Hardware will set the bit, software can clear it with a write of 1b.	Internal hardware events set a sticky bit. Read is from the sticky bit. A write of 1b clears the sticky bit.
Double Buffered	Write when desired and the written value will take effect at the time of the double buffer update point. Reads will return the written value, which is not necessarily the value being currently used to control hardware. 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. Write data is stored into first stage. Read is from the first stage stored data. First stage stored value is transferred to second stage storage at the double buffer update point. Second stage stored value is used to control hardware. Arm/disarm logic may be used for some registers to control the double buffer update point.

Register Access Field	Description	Implementation
Write/Read Status	The value written into this register will control hardware. The read value is determined by hardware.	Write data is stored. Stored value is used to control hardware. Read is from a status signal or some other internal source.

Display Mode Set Sequence

A mode set sequence is the programming sequence that must be followed when enabling or disabling output to a display. There are several different mode set sequences documented in the following sections. The sequence to use depends on which type of port is being enabled or disabled.

Mode Set Triggers

Many display configuration changes can be performed on the fly, but some display configuration changes can only be done as part of a mode set. Those changes are considered to have triggered a mode set. Most triggers will just affect a single pipe and port, but some can affect multiple pipes and ports.

The types of mode sets here are separated into "port" and "pipe", depending on what logic needs to be disabled and re-enabled.

Port mode set definition:

- A port mode set requires the complete disable and enable sequences, involving the port and the pipe(s) attached to the port and the planes attached to the pipe(s).

Pipe mode set definition:

- In DisplayPort SST, FDI (CRT DAC), HDMI, and DVI modes a pipe mode set requires the complete disable and enable sequences, involving the pipe and the planes and port attached to the pipe.
- In DisplayPort MST mode a pipe mode set requires a partial disable and enable sequence (stream deletion and addition), involving the pipe and the planes attached to the pipe, but not the port.

The following table lists the triggering configuration changes, any qualifiers, and the type of mode set required.

Mode Set Triggers

Register	Bit Field	Change Trigger	Qualifier	Type of Mode Set	Notes
LCPLL_CTL	PLL Disable	Enable to disable	Any CPU display feature is enabled	Both for all ports and pipes, and also turn off all CPU display features	
LCPLL_CTL	Reference Select	Any change	LCPLL_CTL is enabled	Both for all ports and pipes, and also turn off all CPU display features	

Register	Bit Field	Change Trigger	Qualifier	Type of Mode Set	Notes
LCPLL_CTL	CD frequency select	Any change	Any CPU display feature is enabled	Both for all ports and pipes, and also turn off all CPU display features	
LCPLL_CTL	CD* clock disables	Enable to disable	Any CPU display feature is enabled	Both for all ports and pipes, and also turn off all CPU display features	
SPLL_CTL	PLL Enable	Enable to disable	Any PORT_CLK_SEL points to SPLL	Both for all ports and pipes that are using SPLL	
SPLL_CTL	Reference Select	Any change	SPLL enabled	Both for all ports and pipes that are using SPLL	
SPLL_CTL	Frequency select	Any change	Any PORT_CLK_SEL points to SPLL and the attached DDI is done with training	Both for all ports and pipes that are using SPLL	
WRPLL_CTL	PLL Enable	On to off	Any PORT_CLK_SEL points to WRPLL	Both for all ports and pipes that are using WRPLL	
WRPLL_CTL	Reference Select	Any change	WRPLL enabled	Both for all ports and pipes that are using WRPLL	
WRPLL_CTL	Divider fields	Any change	Any PORT_CLK_SEL points to WRPLL and the attached DDI is done with training	Both for all ports and pipes that are using	

Register	Bit Field	Change Trigger	Qualifier	Type of Mode Set	Notes
				WRPLL	
PORT_CLK_SEL	Port clock select	Any change	DDI_BUF_CTL or DP_TP_CTL is enabled or any PIPE_CLK_SEL is pointing to this port	Both for port and any pipe attached to this port	
PIPE_CLK_SEL	Pipe clock select	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled	Pipe	
PWR_WELL_CTL<1,2,3,4>	Power Well Request	Enable to disable	Anything in power well is enabled	Ports B, C, D, E; Pipe B and C; Pipe A if not using the always on path to EDP	
ARB_CTL	Any field	Any change	Any planes enabled (PRI_CTL_* Primary_Plane_Enable, SPR_CTL_* Sprite_Enable, CUR_CTL_* Cursor_Mode_Select)	All pipes	
ARB_CTL2	Any field	Any change	Any planes enabled (PRI_CTL_* Primary_Plane_Enable, SPR_CTL_* Sprite_Enable, CUR_CTL_* Cursor_Mode_Select)	All pipes	
PIPE_HTOTAL, PIPE_HBLANK, PIPE_HSYNC, PIPE_VTOTAL, PIPE_VBLANK, PIPE_VSYNC, PIPE_VSYNCSHIFT, PIPE_MULT	Any field	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled	Pipe	
PIPE_SRCSZ	Any field	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled and panel fitter is not enabled	Pipe	Source size can only change on the fly if panel fitter is enabled
DATAM, DATAN, LINKM, LINKN <1_A,1_B,1_C>	Any field	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled in DP SST, DP MST, or FDI mode.	Pipe	Not allowed to change on the fly for these non-EDP pipes.
DATAM, DATAN, LINKM, LINKN <1_EDP>	Any field	Any change	DP mode and PIPE_CONF enabled and PIPE_CONF Refresh_Rate_Switch = 0 (DRRS	Pipe	EDP can only change the M/N that is

Register	Bit Field	Change Trigger	Qualifier	Type of Mode Set	Notes
			using M/N1)		not currently being used by DRRS.
DATAM, DATAN, LINKM, LINKN <2_EDP>	Any field	Any change	DP mode and PIPE_CONF enabled and PIPE_CONF Refresh_Rate_Switch = 1 (DRRS using M/N2)	Pipe	EDP can only change the M/N that is not currently being used by DRRS.
PIPE_DDI_FUNC_CTL	Pipe DDI Function Enable	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled	Pipe	
PIPE_DDI_FUNC_CTL	DDI Select	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled	Pipe	
PIPE_DDI_FUNC_CTL	DDI Mode Select	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled	Pipe	
PIPE_DDI_FUNC_CTL	Bits per color	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled	Pipe	
PIPE_DDI_FUNC_CTL	EDP Input Select	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled	Pipe	
PIPE_DDI_FUNC_CTL	BFI enable	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled	Pipe	
PIPE_DDI_FUNC_CTL	DP port width	Any change	PIPE_CONF or PIPE_DDI_FUNC_CTL or DP_TP_CTL are enabled	Pipe and port	
PIPE_CONF	Pipe enable	Any change	DP SST: PIPE_CONF or PIPE_DDI_FUNC_CTL or DP_TP_CTL are enabled DP MST: PIPE_CONF or PIPE_DDI_FUNC_CTL are enabled		
PIPE_CONF	Interlaced mode	Any change	PIPE_CONF enabled	Pipe	
PF_CTRL	Pipe select	Any change	PF_CTRL is enabled	Pipe	
PF_CTRL	7x5 reconfig enable	Any change	PF_CTRL is enabled	Pipe	
DP_TP_CTL	Transport enable	Enable to disable	DP_TP_CTL enabled	Port	

Register	Bit Field	Change Trigger	Qualifier	Type of Mode Set	Notes
DP_TP_CTL	Transport mode select	Any change	PIPE_CONFG or PIPE_DDI_FUNC_CTL or DP_TP_CTL enabled	Both for port and any pipe attached to this port	
DP_TP_CTL	Enhanced framing enable	Any change	DP_TP_CTL enabled	Port	
DP_TP_CTL	Link training	Normal to training pattern	DP_TP_CTL enabled	Port	
DP_TP_CTL	Scrambling disable	Any change	DP_TP_CTL enabled	Port	
DP_TP_CTL	Alternate SR enable	Any change	DP_TP_CTL enabled	Port	
DDI_BUF_CTL	DDI buffer enable	Any change	DP_TP_CTL or DDI_BUF_CTL enabled	Port	
DDI_BUF_CTL	Port reversal	Any change	DP_TP_CTL or DDI_BUF_CTL enabled	Port	
DDI_BUF_CTL	DDIA lane capability control	Any change	DDI_BUF_CTL_A or DDI_BUF_CTL_E enabled	Port for both DDI A and DDI E	
DDI_BUF_CTL	DP port width selection	Any change	DDI_BUF_CTL enabled	Port	
DDI_BUF_TRANS	Any field	Any change	DDI_BUF_CTL enabled	Port	
PCH HTOTAL, HBLANK, HSYNC, VTOTAL, VBLANK, VSYNC, VSYNCSHIFT	Any field	Any change	TRANS_CONF or DAC_CTL are enabled	Port	
PCH DAC_CTL	Port enable	Any change	TRANS_CONF or DAC_CTL are enabled	Port	
PCH TRANS_CONF	Transcoder enable	Any change	TRANS_CONF or DAC_CTL are enabled	Port	
PCH TRANS_CONF	Interlaced mode	Any change	TRANS_CONF or DAC_CTL are enabled	Port	
PCH FDI_RX_CTL	FDI Rx enable	Enable to disable	FDI_RX_CTL enabled	Port	
PCH FDI_RX_CTL	Port width	Any	FDI_RX_CTL enabled	Port	

Register	Bit Field	Change Trigger	Qualifier	Type of Mode Set	Notes
	selection	change			
PCH FDI_RX_CTL	Polarity reversal	Any change	FDI_RX_CTL enabled	Port	
PCH FDI_RX_CTL	Link reversal stap override	Any change	FDI_RX_CTL enabled	Port	
PCH FDI_RX_CTL	FDI PLL enable	Enable to disable	FDI_RX_CTL Rawclk_to_PCDCLK_selection = 1	Port	Can only be changed at specific point in mode set sequence.
PCH FDI_RX_CTL	Scrambling disable	Any change	FDI_RX_CTL enabled	Port	
PCH FDI_RX_CTL	Enhanced framing enable	Any change	FDI_RX_CTL enabled	Port	
PCH FDI_RX_CTL	Rawclk to PCDCLK selection	Any change	Any PCH display feature enabled	Port	Can only be changed at specific point in mode set sequence.
PCH FDI_RX_MISC	Any field	Any change	FDI_RX_CTL enabled	Port	

Sequence for CRT Port

FDI Programming for Boot With Internal Graphics Disabled
This initial setup must be run once during boot to configure the PCH FDI I/O to save power when internal graphics will not be enabled. Follow iCLKIP Register Information and Programming Guide, Programming of FDI Reference and mPHY, Sequence to configure PCH FDI I/O.

Enable Sequence for CRT Port
DDIA Lane Capability Control must be configured prior to enabling any ports or port clocks DDI_BUF_TRANS must be configured prior to enabling DDI_BUF_CTL Workaround: Program FDI_RX_MISC TP1 to TP2 time with the default value before enabling the FDI receiver. Workaround: Program FDI_RX_MISC FDI Delay to 90h before enabling the FDI receiver. Workaround: Program DISPIO_CR_TX_BMU_CR4 [24:12] = 1_1010_0010_1000b before enabling DDI_BUF_CTL. Enable Power Well 1. If any required resource is in the power well a. Enable power well b. Wait for power well to complete enabling (read status bit and timeout after 20 µs) Setup and Enable CPU PLL and SSC reference 2. PCH SSC reference a. Follow iCLKIP Register Information and Programming Guide, Programming of FDI Reference and mPHY, Sequence to enable CLKOUT_DP for FDI usage and configure PCH FDI I/O. b. Wait 20 µs for DMI latency

Enable Sequence for CRT Port
3. Configure and enable desired CPU Display PLL, either the SPLL or WRPLL, wait 20 µs for warmup Enable and Train FDI 4. [Exclude DevLPT:H:A] Program FDI_RX_MISC FDI RX Pwrdn Lane1 to 10b and FDI RX Pwrdn Lane0 to 10b. 5. Enable PCH FDI Receiver PLL, wait 200 µs for warmup plus 20 µs DMI latency 6. Switch from Rawclk to PCDclk in FDI Receiver 7. Configure Port Clock Select to direct the CPU Display PLL to the port 8. Configure and enable DP_TP_CTL with auto training selected 9. Configure and enable DDI_BUF_CTL 10. Wait >518 µs for buffers to enable before starting training 11. Program PCH FDI Receiver TU size same as Transmitter TU size 12. Enable PCH FDI Receiver with auto training enabled 13. Wait 30 µS for FDI receiver lane calibration 14. [Exclude DevLPT:H:A] Program FDI_RX_MISC FDI RX Pwrdn Lane1 to 00b and FDI RX Pwrdn Lane0 to 00b. 15. Wait 5 µs for FDI auto training time 16. Read DP_TP_STATUS register for auto train done • If not done, see note on FDI training failure Enable Planes and Pipe 17. Configure Pipe Clock Select to direct the Port clock to the Pipe 18. Configure and enable planes (VGA or hi-res). This can be done later if desired. • A workaround is needed before enabling planes (primary, sprite, cursor) when transitioning from no pipes or a single pipe enabled to multiple pipes enabled. Refer to the Notes below for details. 19. If VGA - Clear VGA I/O register SR01 bit 5 20. Enable panel fitter if needed (must be enabled for VGA) 21. Configure pipe timings, M/N/TU, and other pipe settings 22. Configure and enable PIPE_DDI_FUNC_CTL 23. Configure and enable PIPE_CONF 24. Workaround: If only a single pipe was enabled before the previous step, then wait for one vertical blank on that pipe. Setup and Enable PCH Pixel Clock 25. Follow iCLKIP Register Information and Programming Guide, Programming of VGA Pixel Clock, Sequence to

Enable Sequence for CRT Port
enable VGA pixel clock. Enable Transcoder 26. Configure PCH transcoder timings and other transcoder settings (should match CPU settings) 27. Workaround: Set timing override 0xF0064 bit 31 = 1. 28. Enable PCH TRANS_CONF Enable CRT Port 29. Enable CRT port
Notes
Workaround: When transitioning from no pipes or a single pipe enabled to multiple pipes enabled, the plane (primary, sprite, cursor) enabling needs to be adjusted. When enabling planes after pipe is enabled: Enable the second pipe (skipping the plane enabling), wait for at least two vertical blank starts on the first pipe, then enable the planes for the second pipe. When enabling planes before pipe is enabled: Set register 45280h bits 2:1 to 11b, wait for at least one vertical blank start on the first pipe, enable the planes for the second pipe, enable the second pipe as usual, then restore 45280h bits 2:1. For CRT, a pipe is considered to be enabled once the enable sequence has completed the step that enables TRANS_CONF. When a FDI training failure is detected, retry training at the next available voltage swing and pre-emphasis setting. Each setting should be tried at least twice before failing the mode set. To retry FDI training, follow the Disable Sequence steps to Disable FDI, but skip the steps related to clocks and PLLs (16, 19, and 20), then follow the Enable Sequence starting at the step that DP_TP_CTL is configured and enabled. If the mode set fails, follow the disable sequence to disable everything that had been enabled.
Disable Sequence for CRT Port
Disable Planes and Pipe 1. If VGA

Enable Sequence for CRT Port

a. Set VGA I/O register SR01 bit 5 for screen off
b. Wait for 100 µs

2. Disable planes (VGA or hires)
3. Disable pipe in PIPE_CONF
4. Wait for pipe off status in PIPE_CONF, timeout after two frame times
5. Disable PIPE_DDI_FUNC_CTL with DDI_Select set to None
   - HSW-X0 Workaround: After disabling PIPE_DDI_FUNC_CTL, re-enable PIPE_DDI_FUNC_CTL with DDI_Select set to None then again disable with DDI_Select set to None
6. Disable panel fitter
7. Configure Pipe Clock Select to direct no clock to the pipe

Disable Port

8. Disable CRT port DAC_CTL

Disable Transcoder

9. Disable PCH TRANS_CONF
10. Wait for transcoder off status in PCH TRANS_CONF, timeout after two frame times
11. Workaround: Clear timing override 0xF0064 bit 31 = 0.

Disable PCH Pixel Clock

12. Follow iCLKIP Register Information and Programming Guide, Programming of VGA Pixel Clock, Sequence to disable VGA pixel clock.

Disable FDI

13. Disable DDI_BUF_CTL
    - Workaround: Disable PCH FDI Receiver before disabling DDI_BUF_CTL.
14. Disable DP_TP_CTL (do not set port to idle when disabling)
15. Wait 8 µs or poll on DDI_BUF_CTL Idle Status for buffers to return to idle
16. Configure Port Clock Select to direct no clock to the port
17. Disable PCH FDI Receiver
18. [Exclude DevLPT:H:A] Program FDI_RX_MISC FDI RX Pwrdn Lane1 to 10b and FDI RX Pwrdn Lane0 to 10b.
19. Switch from PCDCLK to Rawclk in PCH FDI Receiver

Enable Sequence for CRT Port
20. Disable PCH FDI Receiver PLL Disable CPU PLL and SSC reference 21. If CPU Display PLL no longer needed, disable CPU Display PLL (this applies to the SPLL and WRPLLs, not LCPLL) 22. If SSC clock reference no longer needed, follow iCLKIP Register Information and Programming Guide, Programming of FDI Reference and mPHY, Sequence to disable CLKOUT_DP for FDI usage. Disable Power Well 23. If no required resource is in the power well - Disable power well

Sequence for DisplayPort

This applies to both MST (multi-stream) and SST (single stream) modes of operation.

Enable sequence for DisplayPort
DDIA Lane Capability Control must be configured prior to enabling any ports or port clocks DDI_BUF_TRANS must be configured prior to enabling DDI_BUF_CTL Enable Power Well 1. If any required resource is in the power well a. Enable power well b. Wait for power well to complete enabling (read status bit and timeout after 20 µs) Enable Panel Power 2. If panel power sequencing is required a. Enable panel power sequencing b. Wait for panel power sequencing to reach the enabled state c. Retrieve panel configuration using AUX channel Setup and Enable CPU PLL and SSC reference 3. If SSC reference is needed a. Follow iCLKIP Register Information and Programming Guide, Programming of CLKOUT_DP, Sequence to enable CLKOUT_DP. b. Wait 20 µs for DMI latency 4. Configure and enable desired CPU Display PLL, wait 20 µs for warmup Enable and Train DisplayPort 5. Configure Port Clock Select to direct the CPU Display PLL to the port 6. Configure and enable DP_TP_CTL with link training pattern 1 selected

Enable sequence for DisplayPort
7. Configure and enable DDI_BUF_CTL 8. Wait >518 µs for buffers to enable before starting training or allow for longer time in TP1 before software timeout 9. Follow DisplayPort specification training sequence (see note on failure handling) 10. Set DP_TP_CTL link training to Idle Pattern, wait for 5 idle patterns (DP_TP_STATUS Min_Idles_Sent) (timeout after 800 µs) 11. Set DP_TP_CTL link training to Normal, skip if eDP (DDI A) Enable Planes and Pipe(s) **(repeat to add multiple pipes on a single port for multi-streaming)** 12. If DisplayPort multi-stream - use AUX to program receiver VC Payload ID table to add stream 13. Configure Pipe Clock Select to direct the Port clock to the Pipe 14. Configure and enable planes (VGA or hi-res). This can be done later if desired. • A workaround is needed before enabling planes (primary, sprite, cursor) when transitioning from no pipes or a single pipe enabled to multiple pipes enabled. Refer to the Notes below for details. 15. If VGA - Clear VGA I/O register SR01 bit 5 16. Enable panel fitter if needed (must be enabled for VGA) 17. Configure pipe timings, M/N/TU/VC payload size, and other pipe settings 18. Configure and enable PIPE_DDI_FUNC_CTL 19. If DisplayPort multi-stream - Enable pipe VC payload allocation in PIPE_DDI_FUNC_CTL 20. If DisplayPort multi-stream - Wait for ACT sent status in DP_TP_STATUS and receiver DPCD (timeout after >410us) 21. Configure and enable PIPE_CONF 22. If eDP (DDI A), set DP_TP_CTL link training to Normal 23. If panel power sequencing is required for the display attached to this pipe - Enable panel backlight SRD and/or Audio can be enabled after everything is complete. Follow audio enable sequence documented in the audio registers section.
Notes
Workaround: When transitioning from no pipes or a single pipe enabled to multiple pipes enabled, the plane (primary, sprite, cursor) enabling needs to be adjusted. When enabling planes after pipe is enabled: Enable the second pipe (skipping the plane enabling), wait for at least two vertical blank starts on the first pipe, then enable the planes for the second pipe.

Enable sequence for DisplayPort
When enabling planes before pipe is enabled: Set register 45280h bits 2:1 to 11b, wait for at least one vertical blank start on the first pipe, enable the planes for the second pipe, enable the second pipe as usual, then restore 45280h bits 2:1. For embedded DisplayPort, a pipe is considered to be enabled once the enable sequence has completed the step that sets link training to Normal. For DisplayPort, a pipe is considered to be enabled once the enable sequence has completed the step that enables PIPE_CONF. When a DisplayPort training failure is detected, disable DP_TP_CTL, reconfigure voltage swing and emphasis, then restart training. Retries should iterate through the available voltage swing and emphasis settings. Each setting should be tried at least twice before failing mode set. If the mode set fails, follow the disable sequence to disable everything that had been enabled. If DisplayPort frequency change is necessary during link training, follow the disable sequence steps to disable port and PLL, then change PLL frequency, then follow the enable sequence steps to enable PLL and port and train link.
Synchronizing timing of multiple pipes using DisplayPort
This should result in the synchronized pipes running with closely aligned vertical and horizontal timings. 1. Follow enable sequence for each pipe and port that will be synchronized all the way up to configuring and enabling PIPE_DDI_FUNC_CTL. Do not enable PIPE_CONF. • Use the same PLL and identical pipe and port configurations for all pipes that are to be synchronized. 2. Complete the modified enable sequence for all the pipes and ports that will be synchronized, except do not enable PIPE_CONF. 3. Wait 150uS. 4. Enable PIPE_CONF for the pipes that will be synchronized. • The MMIO register writes to enable PIPE_CONF must be done as close as possible to each other so that all pipes will begin running at the same time. 5. Set link training to normal pixel output for the ports that will be synchronized. Software may still need to add extra synchronizing delays to ensure updates to plane and pipe registers will take place in the same frame. For example: If pipe A and pipe B are synchronized together and software needs the surface addresses for primary plane A and primary plane B to update at the same time, software should wait for vertical blank before writing the surface address registers for both planes, otherwise there is a possibility that the writes could be split across a vertical blank such that one plane would update on the current vertical blank and the other plane would update on

Enable sequence for DisplayPort
the next vertical blank.
Adding streams for multi-stream
Repeat the steps to enable planes and pipes from the enable sequence to add each additional stream.
Disable sequence for DisplayPort

SRD and Audio must be disabled first.

Follow audio disable sequence documented in the audio registers section.

1. If panel power sequencing is required - Disable panel backlight

Disable Planes and Pipe(s) **(repeat to remove multiple pipes from a single port for multi-streaming)**

2. If VGA
    a. Set VGA I/O register SR01 bit 5 for screen off
    b. Wait for 100 µs
3. Disable planes (VGA or hires)
4. Disable pipe in PIPE_CONF
5. Wait for pipe off status in PIPE_CONF, timeout after two frame times
6. If DisplayPort multistream - use AUX to program receiver VC Payload ID table to delete stream
7. If done with this VC payload
    a. Disable VC payload allocation in PIPE_DDI_FUNC_CTL
    b. Wait for ACT sent status in DP_TP_STATUS and receiver DPCD
8. Disable PIPE_DDI_FUNC_CTL with DDI_Select set to None
    - HSW-X0 Workaround: After disabling PIPE_DDI_FUNC_CTL, re-enable PIPE_DDI_FUNC_CTL with DDI_Select set to None then again disable with DDI_Select set to None
9. Disable panel fitter
10. Configure Pipe Clock Select to direct no clock to the pipe

Disable Port (all pipes and VC payloads on this port must already be disabled)

11. Disable DDI_BUF_CTL
12. Disable DP_TP_CTL (do not set port to idle when disabling)
13. Wait 8 µs or poll on DDI_BUF_CTL Idle Status for buffers to return to idle
14. If panel power sequencing is required - Disable panel power sequencing
15. Configure Port Clock Select to direct no clock to the port

Enable sequence for DisplayPort
Disable CPU PLL and SSC reference 16. If CPU Display PLL no longer needed, disable CPU Display PLL (this applies to the SPLL and WRPLLs, not LCPLL) 17. If SSC clock reference no longer needed, follow iCLKIP Register Information and Programming Guide, Programming of CLKOUT_DP, Sequence to disable CLKOUT_DP. Disable Power Well 18. If no required resource is in the power well - Disable power well
Removing streams for multi-stream
Repeat the steps to disable planes and pipes from the disable sequence for each stream that is deleted.

Sequence for HDMI and DVI

Enable sequence for HDMI and DVI
DDI_BUF_TRANS must be configured prior to enabling DDI_BUF_CTL Enable Power Well 1. If any required resource is in the power well a. Enable power well b. Wait for power well to complete enabling (read status bit and timeout after 20 µs) Setup and Enable CPU PLL and SSC reference 2. If SSC or bent reference is needed a. Follow iCLKIP Register Information and Programming Guide, Programming of CLKOUT_DP, Sequence to enable CLKOUT_DP. b. Wait 20 µs for DMI latency 3. Configure and enable desired CPU Display PLL, wait 20 µs for warmup 4. Configure Port Clock Select to direct the CPU Display PLL to the port Enable Planes and Pipe 5. Configure Pipe Clock Select to direct the Port clock to the Pipe 6. Configure and enable planes (VGA or hi-res). This can be done later if desired. • A workaround is needed before enabling planes (primary, sprite, cursor) when transitioning from no pipes or a single pipe enabled to multiple pipes enabled. Refer to the Notes below for details. 7. If VGA - Clear VGA I/O register SR01 bit 5 8. Enable panel fitter if needed (must be enabled for VGA) 9. Configure pipe timings and other pipe settings 10. Configure and enable PIPE_DDI_FUNC_CTL 11. Configure and enable PIPE_CONF Enable Port

Enable sequence for HDMI and DVI
12. Configure and enable DDI_BUF_CTL Audio can be enabled after everything is complete by following the audio enable sequence documented in the audio registers section.
Notes
Workaround: When transitioning from no pipes or a single pipe enabled to multiple pipes enabled, the plane (primary, sprite, cursor) enabling needs to be adjusted. When enabling planes after pipe is enabled: Enable the second pipe (skipping the plane enabling), wait for at least two vertical blank starts on the first pipe, then enable the planes for the second pipe. When enabling planes before pipe is enabled: Set register 45280h bits 2:1 to 11b, wait for at least one vertical blank start on the first pipe, enable the planes for the second pipe, enable the second pipe as usual, then restore 45280h bits 2:1. For HDMI and DVI, a pipe is considered to be enabled once the enable sequence has completed the step that enables DDI_BUF_CTL. If the mode set fails, follow the disable sequence to disable everything that had been enabled.
Disable sequence for HDMI and DVI
Audio must be disabled first, following the audio disable sequence documented in the audio registers section. Disable Planes and Pipe 1. If VGA a. Set VGA I/O register SR01 bit 5 for screen off b. Wait for 100 µs 2. Disable planes (VGA or hires) 3. Disable pipe in PIPE_CONF 4. Wait for pipe off status in PIPE_CONF, timeout after two frame times 5. Disable PIPE_DDI_FUNC_CTL with DDI_Select set to None • HSW-X0 Workaround: After disabling PIPE_DDI_FUNC_CTL, re-enable PIPE_DDI_FUNC_CTL with DDI_Select set to None then again disable with DDI_Select set to None 6. Disable panel fitter

Enable sequence for HDMI and DVI
7. Configure Pipe Clock Select to direct no clock to the pipe Disable Port 8. Disable DDI_BUF_CTL 9. Wait 8 µs or poll on DDI_BUF_CTL Idle Status for buffers to return to idle 10. Configure Port Clock Select to direct no clock to the port Disable CPU PLL and SSC reference 11. If CPU Display PLL no longer needed, disable CPU Display PLL (this applies to the SPLL and WRPLLs, not LCPLL) 12. If SSC clock reference no longer needed, follow iCLKIP Register Information and Programming Guide, Programming of CLKOUT_DP, Sequence to disable CLKOUT_DP. Disable Power Well 13. If no required resource is in the power well - Disable power well

Display Sequences for Changing CD Clock Frequency

The CD clock frequency should only be set once after boot or resume, before any display pipe, port, plane, or audio is enabled.

The CD clock frequency availability and supported selections depends on the device type and the Display CDCLK Limit fuse found in register FUSE_STRAP (GTTMMADDR offset 0x42014) bit 24.

If Display CDCLK Limit fuse = 1

- CD clock frequency should be programmed to 450 MHz (900 MHz CD2X clock).

If Display CDCLK Limit fuse = 0

- ULT: CD clock frequency should be programmed to 450 MHz (900 MHz CD2X clock).
- ULX: CD clock frequency should be programmed to the alternate frequency 337.5 MHz (675 MHz CD2X clock).
- Non-ULT and Non-ULX: CD clock frequency can be programmed to 450 MHz (900 MHz CD2X clock) or the alternate frequency 540 MHz (1080 MHz CD2X clock).

Restrictions
The CD clock frequency should only be set once after boot or resume, before any display pipe, port, plane, or audio is enabled. Do not select the alternate CD clock frequency if the Display CDCLK Limit fuse indicates it is not supported. The CD clock frequency can only be changed while the CD clock is running. The CD clock frequency impacts the maximum supported pixel rate and display watermark programming.
Sequence for Changing CD Clock Frequency
1. Change LCPLL_CTL CD clock frequency select (GTTMMADDR offset 0x130040 bits 27:26) to the desired frequency • The frequency change will complete within a few clock cycles. 2. ULX: Inform power controller of the selected frequency a. If selecting 450 MHz CD clock, write GT Driver Mailbox Data0 (GTTMMADDR offset 0x138128) = 0x00000000. If selecting 337.5 MHz CD clock, write GT Driver Mailbox Data0 = 0x00000001. b. Write GT Driver Mailbox Data1 (GTTMMADDR offset 0x13812C) = 0x00000000. c. Write GT Driver Mailbox Interface (GTTMMADDR offset 0x138124) = 0x80000017. 3. Update programming of register fields that are based on CD clock frequency • These updates do not need to happen immediately. They can be delayed to when the functions are

Restrictions
actually used. Program CPU DisplayPort AUX 2X Bit Clock dividers • Register DDI_AUX_CTL_A, field 2X_Bit_Clock_divider (GTTMMADDR offset 0x64010 bits 10:0) • Register SRD_AUX_CTL, field 2X_Bit_Clock_divider (GTTMMADDR offset 0x64810 bits 10:0) • For CD clock 450 MHz, program 2X_Bit_Clock_divider = 225 decimal. • For CD clock 540 MHz, program 2X_Bit_Clock_divider = 270 decimal. • For CD clock 337.5 MHz, program 2X_Bit_Clock_divider = 169 decimal. • Program Display Audio Controller dividers These registers require the display power well to be enabled at the time of programming. The values will be lost when the display power well is disabled. • Register EM4, field MVALUE (0:3:0 HDABAR offset 0x100C bits 17:0) • Register EM5, field NVALUE (0:3:0 HDABAR offset 0x1010 bits 17:0) • For CD clock 450 MHz, program MVALUE = 4 decimal and NVALUE = 75 decimal. • For CD clock 540 MHz, program MVALUE = 4 decimal and NVALUE = 90 decimal. • For CD clock 337.5 MHz, program MVALUE = 16 decimal and NVALUE = 225 decimal. • Register WM_LINETIME, field IPS Line Time (GTTMMADDR offset 0x45270 bitd 24:16) uses CD clock frequency as part of the calculation.

Display Sequences for LCPLL Disabling

These sequences are used to disable the LCPLL when display engine functions will not be used. This may be when internal graphics is disabled through PCI configuration or when internal graphics is temporarily disabled through MMIO.

Disabling LCPLL will stop any clock derived from the LCPLL outputs, including CD clock. The CD clock can be made to run when LCPLL is disabled by switching the CD clock source to Fclk, but that is only supported with the package C8 sequences; as documented in documented in Display Sequences for Package C8.

Most display engine functions will not operate while LCPLL is disabled. This includes the display audio controller, pipes, ports, planes, backlight PWM (if driving PWM from the CPU display utility pin), graphics interrupts (0:2:0 and 0:3:0 line and MSI interrupts), DDI-A Aux controller, and Global Time Code. The display engine timestamp counter TIMESTAMP_CTR 0x44070 will run, but the value cannot be read when CD clock is stopped.

Access to display engine and display audio controller registers is impacted when the LCPLL is disabled and CD clock is stopped.

PCI configuration writes and reads to Bus:Device:Function 0:2:0 and 0:3:0 registers will complete normally.

IO writes to VGA registers will be dropped gracefully and reads will return all zero data.

Memory writes to the 0xA000 to 0xBFFF (VGA memory) range will be dropped gracefully and reads will return all zero data.

MMIO writes to 0:3:0 HDABAR registers will be dropped gracefully and reads will return all zero data.

MMIO writes to 0:2:0 GTMMADR register offsets ranging from 0x40000 to 0x7FFFF will be dropped gracefully and reads will return all zero data.

MMIO writes to 0:2:0 GTMMADR register offsets outside of 0x40000 to 0x7FFFF will complete normally.

Sequence for Display Software to Disable LCPLL

1. Disable all display engine functions using the full mode set disable sequence on all pipes, ports, and planes.
    - Includes display power well, panel power sequencing, backlight PWM (if driving PWM from the CPU display utility pin), Global Time Code, and audio controller.
2. Disable all graphics interrupts in CPU display and PCH display
3. Disable LCPLL
    a. Set LCPLL_CTL PLL_disable to 1b to disable LCPLL.
    b. Poll for LCPLL_CTL PLL_lock = 0b to indicate LCPLL lost lock.
        - Timeout and continue after 1 ms.
4. Disable LCPLL Comp

Sequence for Display Software to Disable LCPLL
a. Set D_COMP COMP_DISABLE to 1b. b. Wait 100 ns for write to complete. c. Poll for D_COMP RCOMP_IN_PROGRESS = 0b. • Timeout and continue after 1 ms. **LCPLL must be re-enabled before any display engine functions or graphics interrupts may be enabled.**
Sequence for Display Software to Enable LCPLL
1. Enable LCPLL Comp a. Set D_COMP COMP_FORCE to 1b and clear D_COMP COMP_DISABLE to 0b. 2. Enable LCPLL a. Clear LCPLL_CTL PLL_disable to 0b to enable LCPLL. b. Poll for LCPLL_CTL PLL_lock = 1b. • Timeout and fail after 5 ms. 3. Display engine functions and graphics interrupts may be enabled.
Registers for Disabling LCPLL
• LCPLL_CTL is LCPLL_CTL_0_2_0_GTTMMADR at GTTMMADDR offset 0x130040 • PLL_disable is LCPLL_CTL bit 31 • PLL_lock is LCPLL_CTL bit 30 • D_COMP is accessed through GT mailbox registers with commands 10h (read) and 11h (write) • RCOMP_IN_PROGRESS is D_COMP bit 9 • COMP_FORCE is D_COMP bit 8 • COMP_DISABLE is D_COMP bit 0

Display Sequences for Package C8+

Package states C8 and greater are power saving states that can be reached when the display LCPLL is disabled and the display engine and other logic is powered off.

To enable package C8+, display software must follow certain programming sequences to completely disable the display and allow C8+ to be entered.

Hardware will dynamically enter and exit package C8+ when allowed, saving and restoring some of the display state.

Hardware wake from C8+ on PCH Display Interrupt, due to DDI B, DDI C, or DDI D hotplug detection is supported.

Hardware wake from C8+ on DDI A (eDP) hotplug detection is not supported.

Registers used by display software for Package C8+

• LCPLL_CTL is LCPLL_CTL_0_2_0_GTTMMADR at GTTMMADDR offset 0x130040
  • PLL_disable is LCPLL_CTL bit 31
  • PLL_lock is LCPLL_CTL bit 30
  • Display_power_down_allow is LCPLL_CTL bit 22
  • CD_source_select is LCPLL_CTL bit 21
  • CD_source_switching is LCPLL_CTL bit 20
  • CD_source_fclk is LCPLL_CTL bit 19
• D_COMP is accessed through GT mailbox registers with commands 10h (read) and 11h (write).
  • RCOMP_IN_PROGRESS is D_COMP bit 9
  • COMP_FORCE is D_COMP bit 8
  • COMP_DISABLE is D_COMP bit 0

Sequence for display software to allow Package C8+

Caused by driver policy or O/S request to disable display.
1. Disable everything in display using the full mode set disable sequence on all pipes, ports, and planes.
  • Includes panel power sequencing, backlight, backlight PWM, utility pin, Global Time Code, audio controller, and the display power well.
  • The PCH SSC reference CLKOUT_DP) must be disabled.
2. Disable graphics interrupts
  • Clear all pending graphics interrupts (PCH display interrupt line must be 0 to allow C8+).
  • Disable DDIA hotplug detection (eDP HPD) in CPU display and PCH display and disable and mask the

Registers used by display software for Package C8+
associated interrupt. • If DDIB, DDIC, or DDID hotplug detection is required they may be kept enabled along with the associated interrupts (PCH display interrupt, FDI Int). • Disable and mask off all other interrupts in CPU and PCH display. 3. Save state of display software save/restore registers. • The exact registers depend on driver policy. 4. Switch CDclk source to Fclk a. Set LCPLL_CTL CD_source_select to 1b to switch cdclk to fclk source. b. Poll for LCPLL_CTL CD_source_fclk = 1b to indicate switch to fclk is complete. • Timeout and fail after 1 µs. 5. Disable LCPLL a. Set LCPLL_CTL PLL_disable to 1b to disable LCPLL. b. Poll for LCPLL_CTL PLL_lock = 0b to indicate LCPLL lost lock. • Timeout and continue after 1 ms. c. Set D_COMP COMP_DISABLE to 1b. d. Wait 100 ns for write to complete. e. Poll for D_COMP RCOMP_IN_PROGRESS = 0b. • Timeout and continue after 1 ms. 6. Set LCPLL_CTL Display_power_down_allow to 1b to allow power down.
Sequence for display software to disallow Package C8+
Caused by driver policy, O/S requesting display to be enabled, or PCH Display Interrupt (FDI Int) due to DDIB/C/D hotplug. 1. Read LCPLL_CTL and save value for use in later steps. 2. If saved value for LCPLL_CTL Display_power_down_allow = 1b (power down allowed), clear it to 0b to prevent power down. 3. If saved value for LCPLL_CTL CD_source_select = 1b (CDclk source is Fclk), enable LCPLL and switch source to LCPLL. a. Set D_COMP COMP_FORCE to 1b and clear D_COMP COMP_DISABLE to 0b. b. Clear LCPLL_CTL PLL_disable to 0b to enable LCPLL. c. Poll for LCPLL_CTL PLL_lock = 1b. • Timeout and fail after 5 ms. d. Clear LCPLL_CTL CD_source_select to 0b to switch to LCPLL source. e. Poll for LCPLL_CTL CD_source_fclk = 0b to indicate switching complete. • Timeout and fail after 1 µs. 4. If saved value for LCPLL_CTL Display_power_down_allow = 1b (power down allowed), restore display software save/restore registers. • The exact registers depend on driver policy. 5. If exit was due to interrupt, service the interrupt.

Registers used by display software for Package C8+
6. If exit was due to O/S request to enable displays, enable display using full mode set enable sequence.

North Display Engine Shared Functions

VGA

The VGA Control MMIO register is located here. The VGA I/O registers are located in the VGA Registers document.

VGA_CONTROL

Frame Buffer Compression

FBC_CFB_BASE

FBC_CTL

Intermediate Pixel Storage

IPS enable sequence:

- Pre-requisite: IPS cannot be enabled until after at least one plane has been enabled for at least one vertical blank.

1. Enable Plane
2. Wait for Vblank
3. Enable IPS

IPS disable sequence:

- Pre-requisite: IPS must be disabled while there is still at least one plane enabled.

1. Disable IPS
2. Wait for Vblank
3. Disable Plane

IPS_CTL

IPS_STATUS

Interrupts

Display Engine Interrupt Bit Definition

Audio Codec Interrupt Bit Definition

GT Interrupt Bit Definition

Power Management Interrupt Bit Definition

INTERRUPT

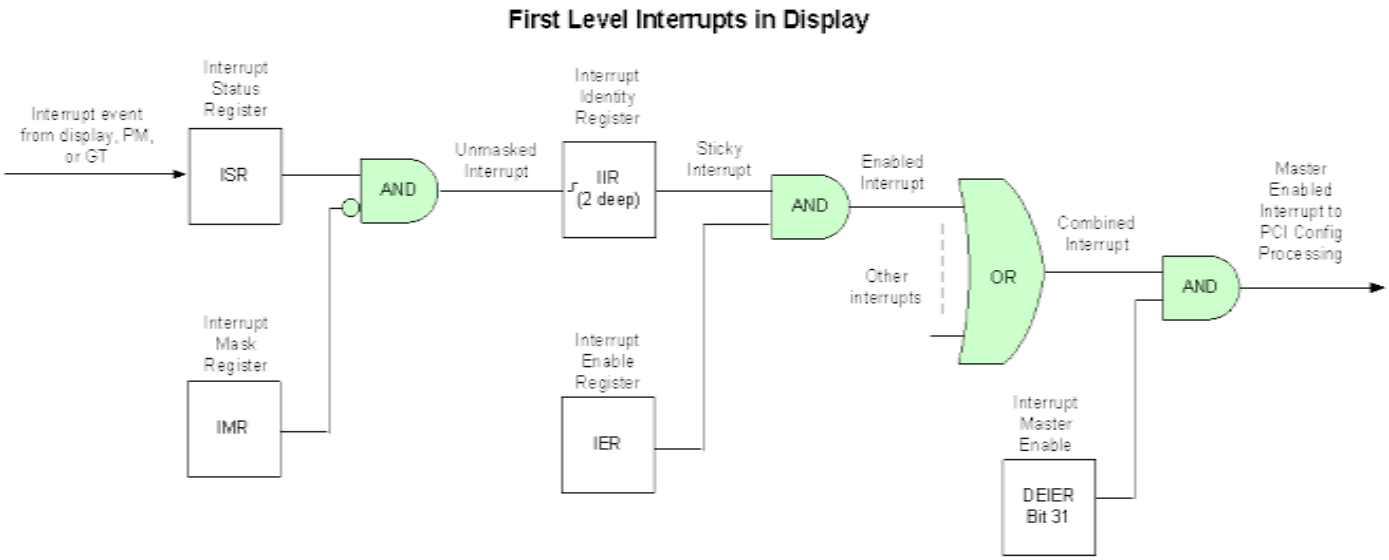
HOTPLUG_CTL

HPD_PULSE_CNT

HPD_FILTER_CNT

ERR_INT

Interrupt Flow



For every first level interrupt bit:

The interrupt event comes in.

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.

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

The live status is not useful for pulse interrupt events due to the short period that the status will be present.

The interrupt event is ANDed with the inverted Interrupt Mask Register (IMR) to create the unmasked interrupt.

Only unmasked interrupts will proceed.

The unmasked interrupt rising edge sets the sticky bit in the Interrupt Indentity Register (IIR).

The IIR can be cleared by writing a 1 to it.

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

The sticky interrupt is ANDed with the Interrupt Enable Register (IER) to create the enabled interrupt.

Only enabled interrupts will proceed.

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

The combined interrupt is ANDed with the Master Interrupt Enable (DEIER Bit 31) to create the master enabled interrupt.

Only a master enabled interrupt will proceed.

The 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 Reset Warn will reset all graphics interrupt logic, causing the master enabled interrupt to de-assert which can cause the MSI or line interrupt to de-assert.

Interrupt Service Routine

1. Disable Master Interrupt Control
   - Clear bit 31 of DE_IER (0x4400c)
   - This is required to prevent missing any interrupts occurring back to back or during the service routine
2. Find the source(s) of the interrupt and clear the Interrupt Identity bits (IIR)
   - Read DE_IIR (0x44008), record which bits are set, then write back 1s to clear the bits that are set
   - Read GT_IIR (0x44018), record which bits are set, then write back 1s to clear the bits that are set
   - Read PM_IIR (0x44028), record which bits are set, then write back 1s to clear the bits that are set
3. Process the interrupt(s) that had bits set in the IIRs
4. Enable Master Interrupt Control
   - Set bit 31 of DE_IER (0x4400c)

Display Engine Render Response

Display Engine Render Response Message Bit Definition

DE_RRMR

Display Arbitration

ARB_CTL

ARB_CTL2

Display Watermarks

The watermark registers are used to control the display to memory request timing. The watermarks must be programmed according to the rules provided in the "Display Watermark Programming" document.

WM_PIPE

WM_LP

WM_LP_SPR

WM_MISC

WM_LINETIME

Watermarks must enable from the bottom up, meaning if WM_LP2 is disabled, WM_LP3 must also be disabled, and if WM_LP1 is disabled, both WM_LP2 and WM_LP3 must also be disabled.

When enabling low power modes with multiple pipes enabled, calculate the watermark values for each pipe separately, then program the worst case in the WM_LP registers.

Display Power Well

When the power well is disabled (powered down), access to any registers in the power will complete, but write data will be dropped and read data will be all zeroes.

The power well enable requests from all sources are logically ORd together to enable the power well, so the power well will only disable after all sources have requested the power well to disable.

PWR_WELL_CTL1

PWR_WELL_CTL2

Display Engine Clocks and Resets

Note: Haswell ULT and ULX clocking are the same, except for the supported CD Clock and DDI frequencies.

LCPLL_CTL

SPLL_CTL

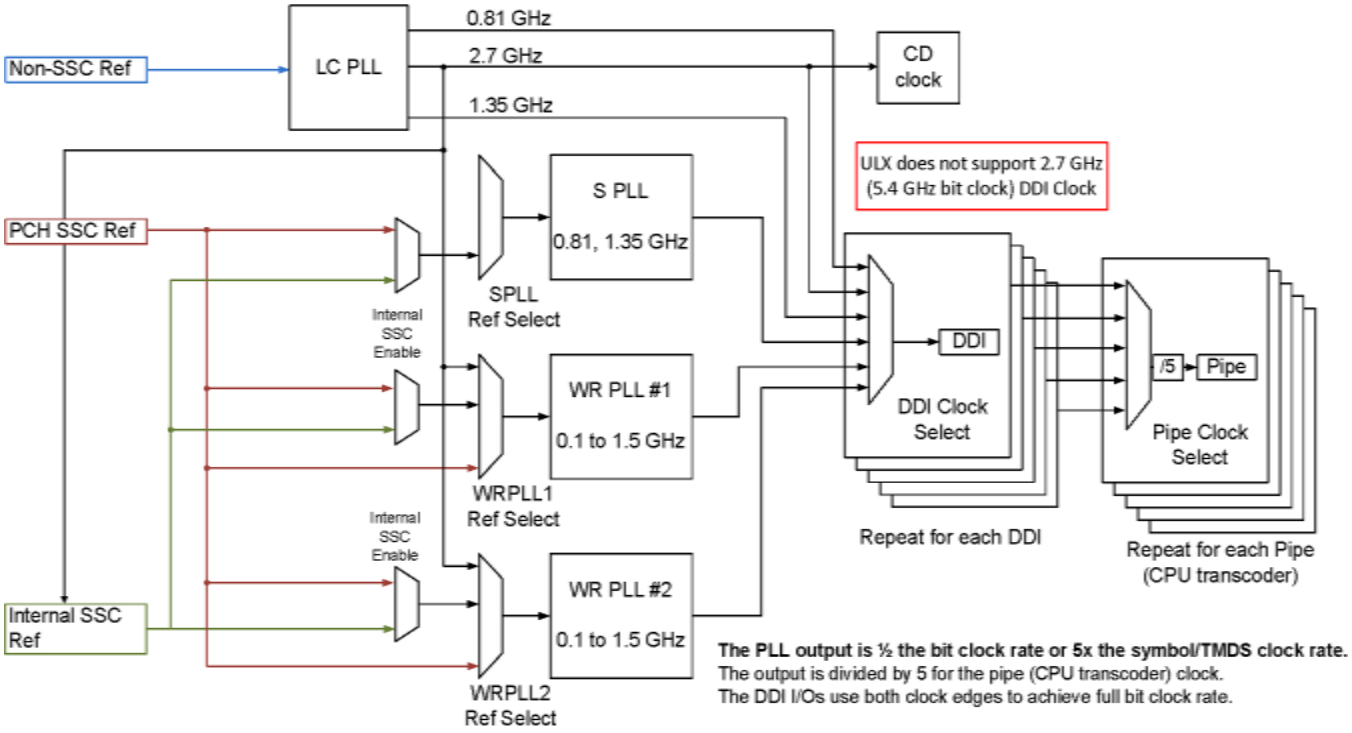
WRPLL_CTL

CDCLK_FREQ

NDE_RSTWRN_OPT

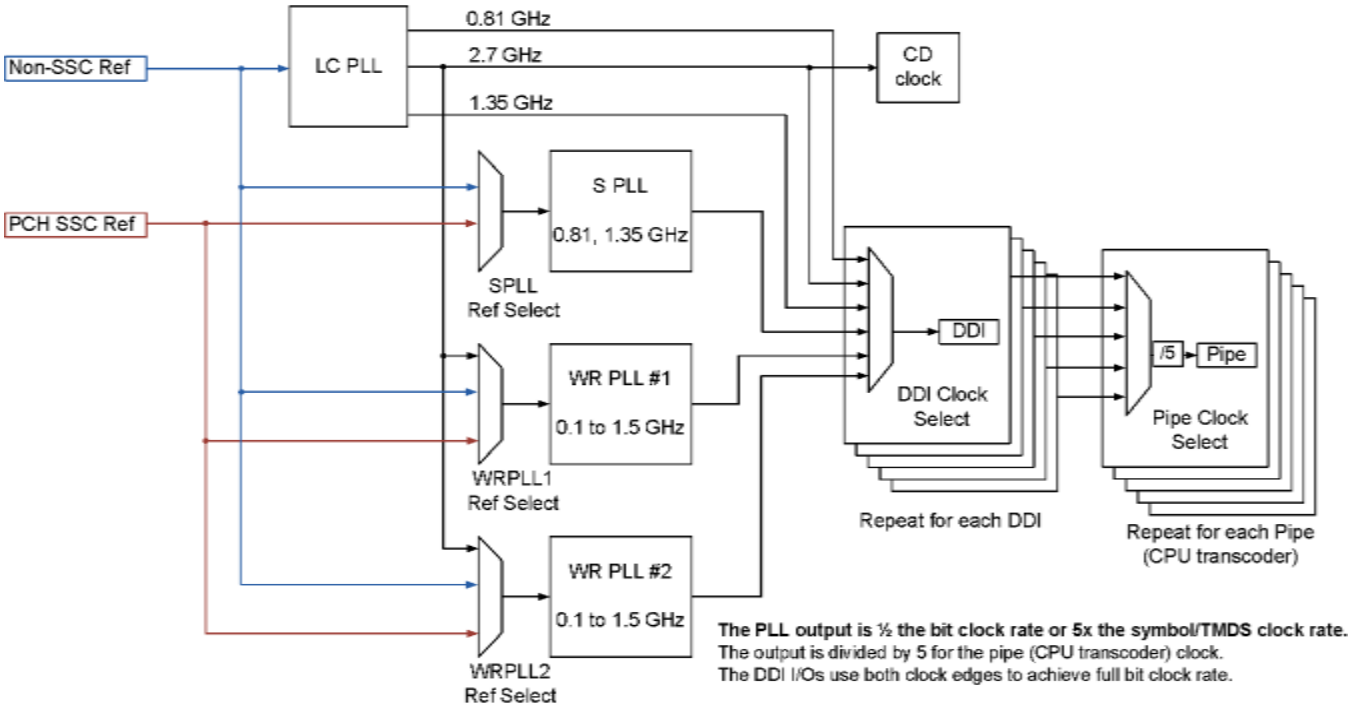
Display Engine Clocks Overview

Supported Display Clock Paths on ULT



The PLL output is ½ the bit clock rate or 5x the symbol/TMDS clock rate.
The output is divided by 5 for the pipe (CPU transcoder) clock.
The DDI I/Os use both clock edges to achieve full bit clock rate.

Supported Display Clock Paths on Non-ULT



The PLL output is ½ the bit clock rate or 5x the symbol/TMDS clock rate.
The output is divided by 5 for the pipe (CPU transcoder) clock.
The DDI I/Os use both clock edges to achieve full bit clock rate.

The display engine clocking structure has multiple reference clocks, PLLs, and clocks. The general flow is from reference to PLL to DDI (port) clock to pipe (CPU transcoder) clock.

Display Engine Clock References

There are multiple display engine clock references.

A single reference may be used by multiple PLLs simultaneously.

	Non-SSC (NSSC) Reference	PCH SSC Reference	Internal SSC Reference
Usage	Non-spread spectrum reference for PLLs	Spread spectrum reference for SPLL and WRPLLs. Can also be used for clock bending; which uses the spread modulator to give a fine adjustment to the frequency to achieve more accurate WRPLL output for certain frequencies.	Spread spectrum reference for SPLL and WRPLLs. Only available on ULT parts.
Input	PCH CLKOUT_DPNS (non-ULT), PCH CLKOUT_CPUNS (ULT)	PCH CLKOUT_DP	LCPLL 2.7 GHz output
Frequency	135 MHz (non-ULT) or 24 MHz (ULT)	Programmable selection between 135 MHz with 0.5% downspread, 135 MHz with 0% spread, and 135Mhz bent +/-~0.5 MHz in ~0.05 MHz steps	135 MHz with 0.5% down spread
Default after reset	Enabled	Disabled	Enabled if fused enabled, disabled and not available for use if fused disabled
Programming	On non-ULT parts may be disabled by software after boot to save power if internal graphics is disabled; as documented in iCLKIP Register Information and Programming Guide - Programming of CLKOUT_DPNS. Not programmable on ULT parts.	Should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence and iCLKIP Register Information and Programming Guide - Programming of CLKOUT_DP.	Not programmable. May be fused enabled or disabled. When fused enabled it changes the reference inputs to the WR PLLs and S PLL.

Display Engine PLLs

There are four display engine PLLs.

The PLL output frequencies are 5x the symbol/TMDS rate (1/2 the bit rate).

The DDI I/Os use both clock edges to achieve full bit clock rate.

The PLL output is divided by 5 to become the symbol/TMDS clock frequency.

A single PLL output may be used by multiple DDI ports simultaneously.

	LC PLL (PLL 1)	S PLL (PLL 2)	WR PLL #1 (PLL 3) and WR PLL #2 (PLL 4)
Usage	Sources for DDI clocks, CD clock, and reference to other PLLs	Source for DDI clocks	Source for DDI clocks
Input	Non-SSC reference	Programmable selection between multiple references, but only SSC references are recommended	Programmable selection between multiple references, but only LCPLL 2.7 GHz output and SSC references are recommended.
Frequency	Simultaneous output of 0.81 GHz, 1.35 GHz, and 2.7 GHz	Programmable selection 0.81 GHz or 1.35 GHz	Programmable selection in the range 0.1 GHz to 1.5 GHz
Default after reset	Enabled	Disabled	Disabled
Programming	May be disabled by software after boot if internal graphics is disabled or for some low power modes; as documented in Display Sequences for LCPLL Disabling and Display Sequences for Package C8. May be enabled and disabled by hardware for some package C states.	Should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence.	Should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence.

DDI Clocks

There is one DDI clock tied to each DDI port.

A single DDI clock output may be used by multiple pipes simultaneously for DisplayPort Multi-streaming.

	DDI clocks A, B, C, D, and E
Usage	DDI port I/O bit clock and symbol/TMDS clock, source for Pipe (CPU transcoder) clocks
Input	Programmable selection between all the PLL outputs**ULX does not support the 2.7 GHz (5.4 GHz bit rate) frequency for DDIs**
Frequency	Input frequency divided by 5 for symbol/TMDS rate and multiplied by 2 for bit rate
Default after reset	Disabled
Programming	Should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence.

Pipe Clocks

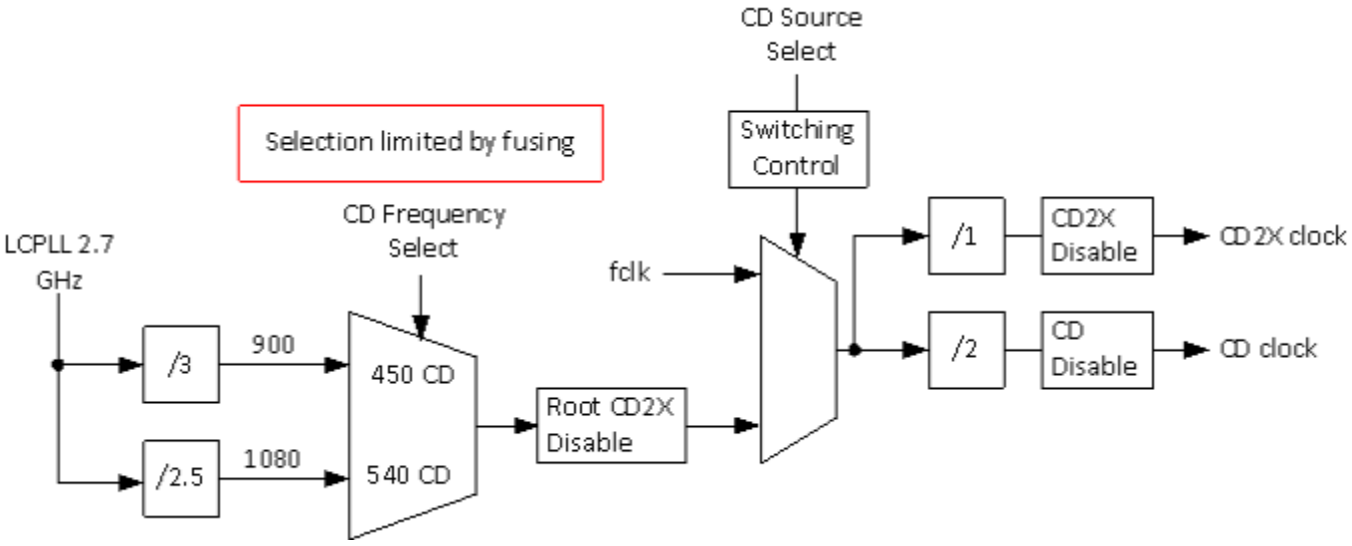
There is one Pipe clock tied to each display pipe (CPU transcoder).

	Pipe clocks A, B, C, and EDP
Usage	Transcoder symbol/TMDS clock
Input	Programmable selection between all the DDI clocks, except pipe EDP always uses DDI A clock
Frequency	PLL output frequency divided by 5
Default after reset	Disabled
Programming	Should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence.

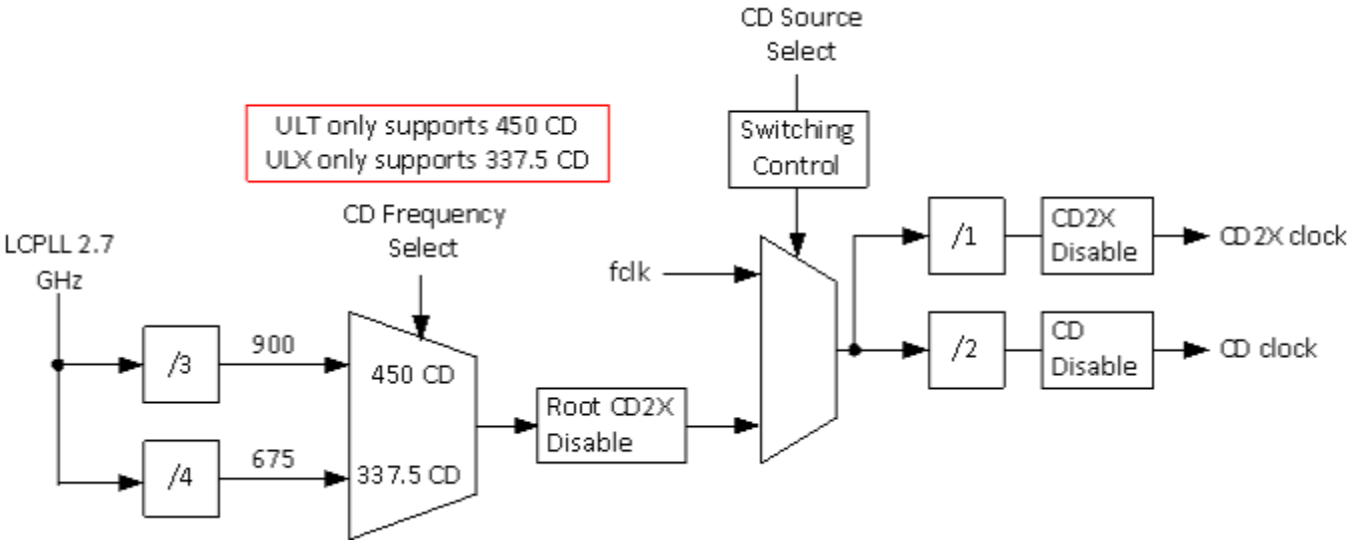
CD Clock

CD clock refers to the Core Display clock which includes the Core Display 1X Clock (CD clock, CDclk, cdclk, CDCLK) and the Core Display 2X Clock (CD2X clock, cd2xclk, CD2XCLK).

CD Clock on Haswell Non-ULT and Non-ULX[1]



CD Clock on Haswell ULT and ULX[1]



[1]The CD clock frequency availability and supported selections depends on the device type and the Display CDCLK Limit fuse found in register FUSE_STRAP (GTTMMADDR offset 0x42014) bit 24.

	CD clock
Usage	Clocking for most display engine functions
Input	Programmable selection between LCPLL 2.7 GHz output and Fclk. Fclk is restricted to use only for package C8 sequences
Frequency	The CD clock frequency availability and supported selections depends on the device type and the Display CDCLK Limit fuse found in register FUSE_STRAP (GTTMMADDR offset 0x42014) bit 24. If Display CDCLK Limit fuse = 1 • CD clock frequency should be programmed to 450 MHz (900 MHz CD2X clock). If Display CDCLK Limit fuse = 0 • ULT: CD clock frequency should be programmed to 450 MHz (900 MHz CD2X clock). • ULX: CD clock frequency should be programmed to the alternate frequency 337.5 MHz (675 MHz CD2X clock). • Non-ULT and Non-ULX: CD clock frequency can be programmed to 450 MHz (900 MHz CD2X clock) or the alternate frequency 540 MHz (1080 MHz CD2X clock).
Default after reset	Enabled with the LCPLL 2.7 GHz source and 450 MHz CD (900 MHz CD2X)
Programming	Frequency switching documented in Display Sequences for Changing CD Clock Frequency. Source switching documented in Display Sequences for Package C8. Has several programmable disable bits, but no programming sequence currently uses them.

South Display Engine Pixel Clock

There is one pixel clock for the south display engine CRT DAC output.

	South Display Engine Pixel Clock
Usage	Clocking for CRT DAC on non-ULT parts. Does not exist on ULT parts.
Input	PCH iCLKIP ICC VGA Pixel Clock
Frequency	Programmable selection between 20 and 180 MHz.
Default after reset	Disabled
Programming	Should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence and iCLKIP Register Information and Programming Guide - Programming of VGA Pixel Clock.

Display Engine Resets

The north and south display engines are reset by PCI Function Level Resets (FLR) and the chip level resets.

A FLR for Bus:Function:Device 0:2:0 will reset the north and south display engines and audio codecs and most of the related MMIO, PCI, and IO configuration registers.

A FLR for Bus:Function:Device 0:3:0 will reset the display audio controller and most of the related MMIO, and PCI configuration registers.

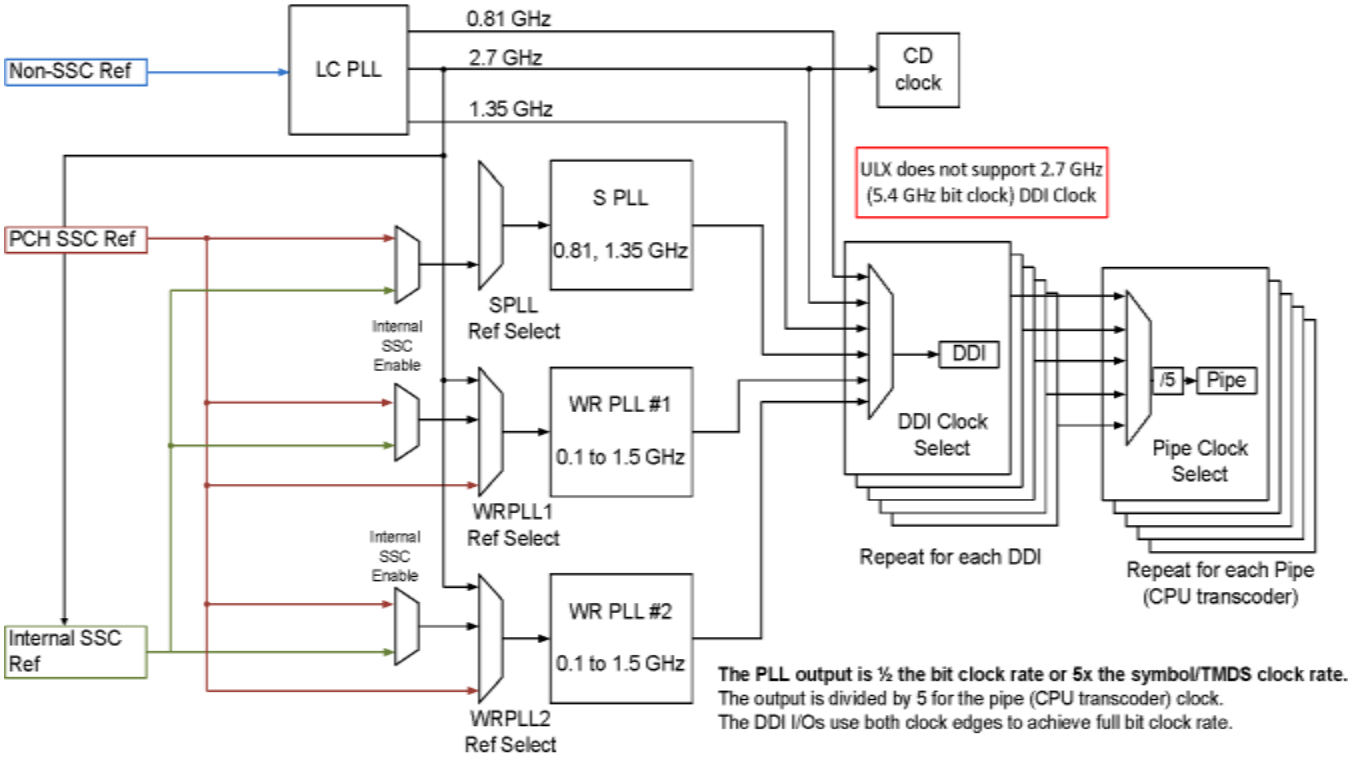
Display configuration registers which are reset by both the chip level reset and by FLR are marked as using the "soft" reset in the programming specification.

Display configuration registers which are reset only by the chip level reset and **not** by FLR are marked as using the "global" reset in the programming specification.

The south display engine will run panel power down sequencing (if configured to do so) before resetting.

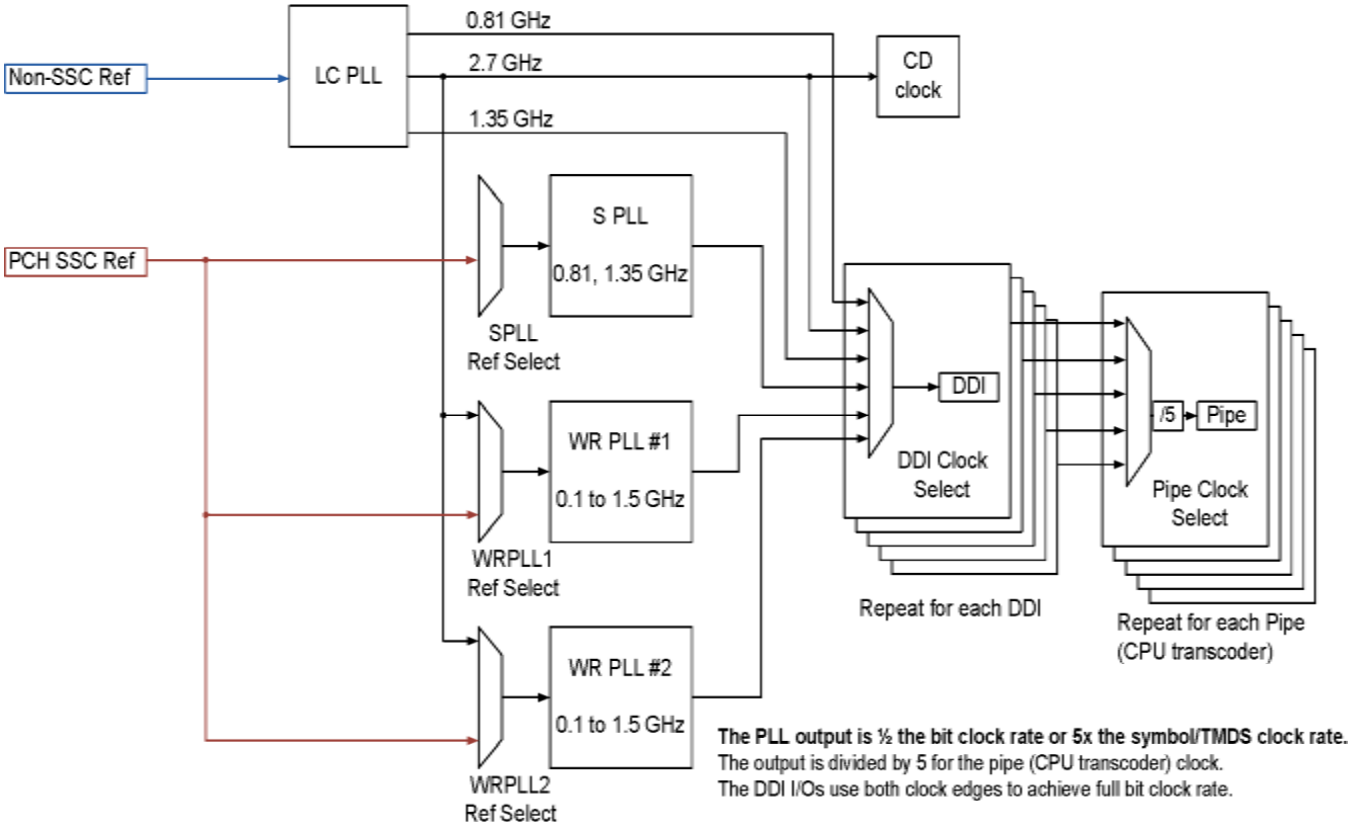
Recommended PLL Selection

Recommended Display Clock Paths on ULT



The PLL output is ½ the bit clock rate or 5x the symbol/TMDS clock rate.
The output is divided by 5 for the pipe (CPU transcoder) clock.
The DDI I/Os use both clock edges to achieve full bit clock rate.

Recommended Display Clock Paths on Non-ULT



The PLL output is ½ the bit clock rate or 5x the symbol/TMDS clock rate.
The output is divided by 5 for the pipe (CPU transcoder) clock.
The DDI I/Os use both clock edges to achieve full bit clock rate.

Table of Recommended PLL Selections per Port Type

Port	Non-SSC 1st preference	Non-SSC 2nd preference	Clock Bending 1st preference	Clock Bending 2nd preference	SSC 1st preference	SSC 2nd preference	SSC 3rd preference
DisplayPort	LC PLL	N/A	N/A	N/A	S PLL	WR PLL #2	WR PLL #1
HDMI/DVI	WR PLL #1	WR PLL #2	WR PLL #1 (PCH SSC only)	WR PLL #2 (PCH SSC only)	N/A	N/A	N/A
FDI	N/A	N/A	N/A	N/A	S PLL (PCH SSC only)	WR PLL #2 (PCH SSC only)	WR PLL #1 (PCH SSC only)

Table of Recommended PLL References

PLL	Supported reference	Recommended for DisplayPort	Recommended for HDMI/DVI	Recommended for FDI (Not for ULT)
LC PLL	Non-SSC	Non-SSC	Not supported	Not supported
S PLL	PCH SSC Internal SSC Non-SSC (Not for ULT)	Internal SSC if fused enabled, else PCH SSC	Not supported	PCH SSC required
WR PLLs	PCH SSC Internal SSC 2.7 GHz LC PLL output Non-SSC (Not for ULT)	Internal SSC if fused enabled, else PCH SSC	Clock bending: PCH SSC Not clock bending: 2.7 GHz LC PLL output	PCH SSC required

Clock bending and SSC:

On ULT parts with internal SSC fused enabled, the PCH SSC reference should be used only for clock bending and the internal SSC reference should be used only for SSC.

On non-ULT parts or ULT parts with internal SSC fused disabled, the PCH SSC reference is used for both clock bending and SSC, so there can be a resource conflict. The recommendation is to select between clock bending and SSC usages on a first come first serve basis, except for FDI which requires SSC. If clock bending is enabled and being used for one or more displays, and FDI must be enabled, disable the

display(s) with the normal disable sequence, switch to SSC, re-enable the display(s) with the normal enable sequence, then go ahead with FDI enabling.

Pseudo-code for Recommended PLL Selection

New display type: D

Clock bit rate: R

Spread spectrum requirement: S

If (D is HDMI or DVI)

    If (WRPLL1 is available) Use WRPLL1 and reference PCH SSC for clock bending, else LCPLL 2700 output

    Else If (WRPLL2 is available) Use WRPLL2 and reference PCH SSC for clock bending, else LCPLL 2700 output

    Else Fail; // No PLL available

Else if (D is DisplayPort)

    If (S == Non-SSC) Use LCPLL;

    Else // SSC

        If (R == 5.4 GHz) Fail; // 5.4 GHz SSC is never available

        Else // 2.7 or 1.62 GHz SSC

            If ((R == 2.7 GHz) and (SPLL 2.7 GHz is available)) Use SPLL and reference SSC

            Else If ((R == 1.62 GHz) and (SPLL 1.62 GHz is available)) Use SPLL and reference SSC

            Else If (WRPLL2 is available) Use WRPLL2 and reference internal SSC if fused enabled, else PCH SSC

            Else If (WRPLL1 is available) Use WRPLL1 and reference internal SSC if fused enabled, else PCH SSC

            Else Fail; // No PLL available;

Else // FDI at 2.7 GHz SSC to PCH CRT DAC

    If (SPLL 2.7 GHz is available) Use SPLL and reference PCH SSC

    Else If (WRPLL2 is available) Use WRPLL2 and reference PCH SSC

    Else If (WRPLL1 is available) Use WRPLL1 and reference PCH SSC

    Else Fail; // No PLL available

Backlight Control

BLC_PWM_CTL

BLC_PWM2_CTL

BLC_PWM_DATA

BLC_MISC_CTL

BLM_HIST_CTL

BLM_HIST_BIN

BLM_HIST_GUARD

Utility Pin

UTIL_PIN_CTL

Color Space Conversion

These registers contain the coefficients of the pipe color space converter.

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

CSC COEFFICIENT FORMAT

CSC_COEFF

CSC_MODE

CSC_PREOFF

CSC_POSTOFF

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

The matrix equations are as follows:

OutputHigh = (CoefficientRU * InputHigh) + (CoefficientGU * InputMedium) + (CoefficientBU * InputLow)

OutputMedium = (CoefficientRY * InputHigh) + (CoefficientGY * InputMedium) + (CoefficientBY * InputLow)

OutputLow = (CoefficientRV * InputHigh) + (CoefficientGV * InputMedium) + (CoefficientBV * InputLow)

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

The input is RGB on high, medium, and low channels respectively.

The output is VYU on high, medium, and low channels respectively.

Program CSC_MODE to put gamma before CSC.

Program the CSC Post-Offsets to +1/2, +1/16, and +1/2 for high, medium, and low channels respectively.

The coefficients and pre and post offsets can be scaled if desired.

	Bt.601		Bt.709	
	Value	Program	Value	Program
RU	0.2990	0x1990	0.21260	0x2D98
GU	0.5870	0x0968	0.71520	0x0B70

	Bt.601		Bt.709	
	Value	Program	Value	Program
BU	0.1140	0x3E98	0.07220	0x3940
RV	-0.1687	0xAAC8	-0.11460	0xBEA8
GV	-0.3313	0x9A98	-0.38540	0x9C58
BV	0.5000	0x0800	0.50000	0x0800
RY	0.5000	0x0800	0.50000	0x0800
GY	-0.4187	0x9D68	-0.45420	0x9E88
BY	-0.0813	0xBA68	-0.04580	0xB5E0

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

The input is VYU on high, medium, and low channels respectively.

The output is RGB on high, medium, and low channels respectively.

Program CSC_MODE to put gamma after CSC.

Program the CSC Pre-Offsets to -1/2, -1/16, and -1/2 for high, medium, and low channels respectively.

The coefficients and pre and post offsets can be scaled if desired.

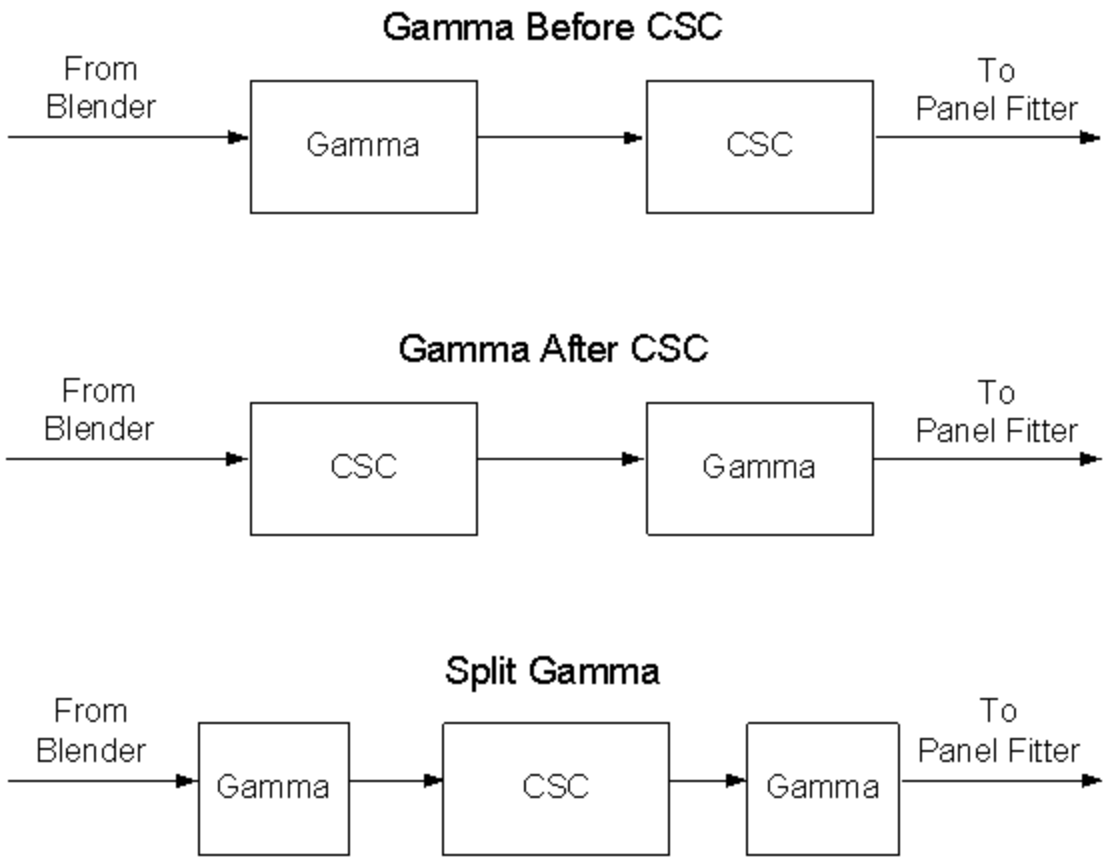
	Bt.601 Reverse		Bt.709 Reverse	
	Value	Program	Value	Program
GY	1.000	0x7800	1.000	0x7800
BY	0.000	0x0000	0.000	0x0000
RY	1.371	0x7AF8	1.574	0x7C98
GU	1.000	0x7800	1.000	0x7800
BU	-0.336	0x9AC0	-0.187	0xABF8
RU	-0.698	0x8B28	-0.468	0x9EF8

	Bt.601 Reverse		Bt.709 Reverse	
	Value	Program	Value	Program
GV	1.000	0x7800	1.000	0x7800
BV	1.732	0x7DD8	1.855	0x7ED8
RV	0.000	0x0000	0.000	0x0000

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

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

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


Gamma Before CSC
From Blender
Gamma
CSC
To Panel Fitter
Gamma After CSC
From Blender
CSC
Gamma
To Panel Fitter
Split Gamma
From Blender
Gamma
CSC
Gamma
To Panel Fitter

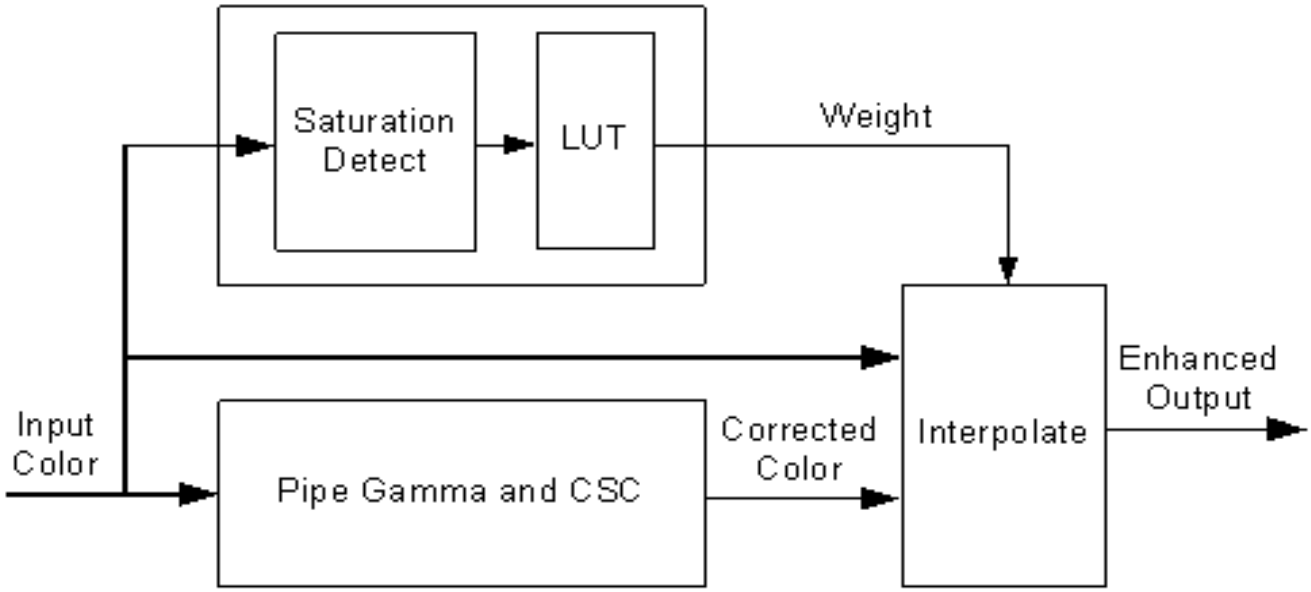
Pipe Color Gamut Enhancement

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

CGE_CTRL

CGE_WEIGHT

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



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

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

Weighting of input and corrected colors

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

Example weight programming

CGE LUT Index	CGE Weight Value Decimal	CGE Weight Value Binary	CGE Weight Percent Input Color	CGE Weight Percent Corrected Color
0 (lowest saturation)	0	00 0000b	0%	100%
1	0	00 0000b	0%	100%
2	0	00 0000b	0%	100%
3	0	00 0000b	0%	100%
4	0	00 0000b	0%	100%
5	0	00 0000b	0%	100%
6	1.6	00 0010b	5%	95%
7	3.2	00 0011b	10%	90%
8	4.8	00 0101b	15%	85%
9	6.4	00 0110b	20%	80%
10	8.64	00 1001b	27%	73%
11	12.8	00 1101b	40%	60%
12	19.2	01 0011b	60%	40%
13	25.6	01 1010b	80%	20%
14	28.8	01 1101b	90%	10%
15	32	10 0000b	100%	0%
16 (highest saturation)	32	10 0000b	100%	0%

Pipe Palette and Gamma

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

PAL_LGC

PAL_PREC_INDEX

PAL_PREC_DATA

PAL_GC_MAX

PAL_EXT_GC_MAX

GAMMA_MODE

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

Restriction: All MMIO write accesses to the palette must be in dwords. MMIO byte or word writes to the palettes are not allowed.

Workaround: Do not access the palette if the pipe is enabled and mode set is not yet complete.

8 bit legacy palette/gamma mode:

This provides a palette mode for indexed pixel data formats (VGA and primary plane 8 bpp) and gamma correction for legacy programming requirements.

All input values are clamped to the 0.0 to 1.0 range before the palette/gamma calculation. It is not recommended to use legacy palette mode with extended range formats.

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

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

10 bit gamma mode:

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

All input values are clamped to the greater than -3.0 and less than 3.0 range before the gamma calculation.

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

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

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

Split gamma mode:

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

First gamma (before CSC):

All input values are clamped to the greater than -3.0 and less than 3.0 range before the gamma calculation.

For input values greater than or equal to 0 and less than 1.0, the input value is used to directly lookup the result value from one of the first 512 gamma entries. The first 512 entries are stored in the precision palette indexes 0 to 511 with 10 bits per color in a 0.10 format with 0 integer and 10 fractional bits.

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

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

Second gamma (after CSC):

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

For input values greater than or equal to 0 and less than 1.0, the input value is used to directly lookup the result value from one of the first 512 gamma entries. The first 512 entries are stored in the precision palette indexes 512 to 1023 with 10 bits per color in a 0.10 format with 0 integer and 10 fractional bits.

12 bit interpolated gamma mode:

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

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

All input values are clamped to the greater than -3.0 and less than 3.0 range before the gamma calculation.

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

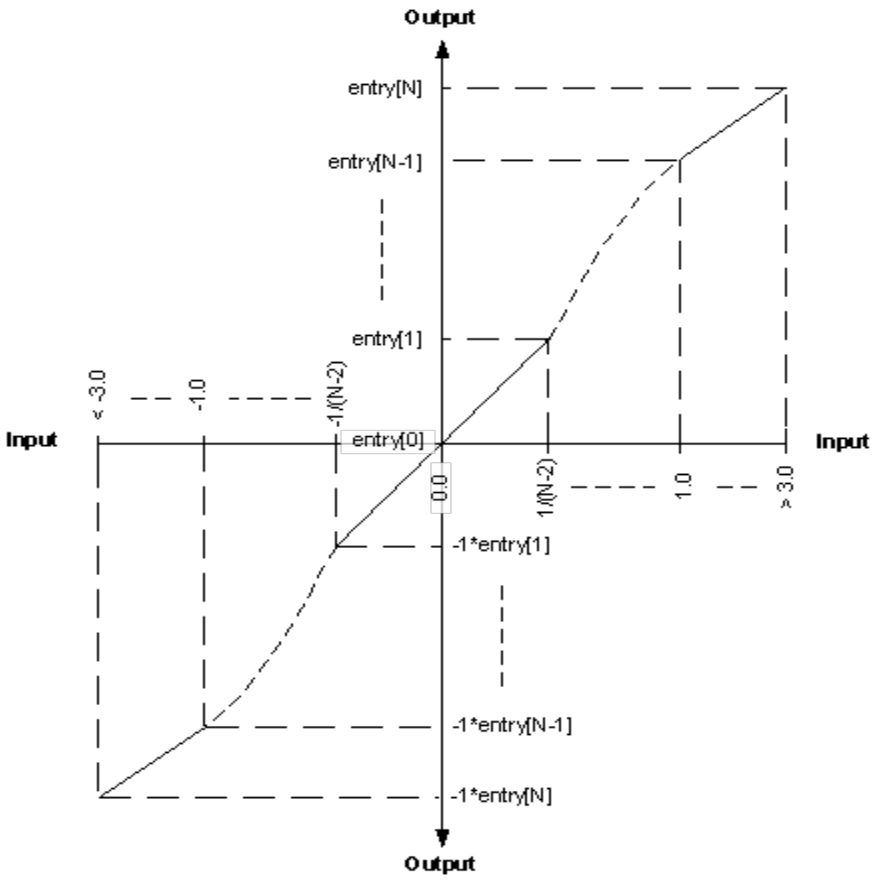
For input values greater than or equal to 1.0 and less than 3.0, the input value is used to linearly interpolate between the 513th and 514th gamma entries to create the result value. The 514th entry is stored in the PAL_EXT_GC_MAX register with 19 bits per color in a 3.16 format with 3 integer and 16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

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

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

Note: If any gamma value to be programmed exceeds the maximum allowable value in the associated gamma register, then the programmed value must be clamped to the maximum allowable value.

Example Pipe Gamma Correction Curve



Software Flags

SWF

GTSCRATCH

North Display Engine Pipe and Port Controls

Pipe Timing

For all timing registers, except Pipe Source Image Size, there is one instance per each pipe timing generator A/B/C/EDP. There is one instant of Pipe Source Image Size per pipe A/B/C.

PIPE_HTOTAL

PIPE_HBLANK

PIPE_HSYNC

PIPE_VTOTAL

PIPE_VBLANK

PIPE_VSYNC

PIPE_SRCSZ

PIPE_VSYNCSHIFT

PIPE_MULT

Pipe M/N Values

These values are used for DisplayPort and FDI.

For dynamic switching between multiple refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. The PIPE_CONF Refresh Rate Switch setting can be changed on the fly and then alternate M/N values will be used in the next frame that is output.

DATAM

DATAN

LINKM

LINKN

Calculation of TU, Data M, and Data N is as follows:

For modes that divide into the link frequency evenly,

Active/TU = Payload/Capacity = Data M/N = dot clock * bytes per pixel / ls_clk * number of lanes

Default value to program TU size is "111111" for TU size of 64.

Calculation of Link M and Link N is as follows:

Link M/N = dot clock / ls_clk

Restriction on clocks and number of lanes:

Number of lanes >= INT(dot clock * bytes per pixel / ls_clk)

cdclk * number of lanes >= dot clock * bytes per pixel

Please note that in the DisplayPort specifcation, dot clock is referred to as strm_clk.

There are restrictions on the Virtual Channel (VC) payload size in DisplayPort MST mode.

In a x1 lane config, each pipe needs to have a VC payload size that is a multiple of 4.

In a x2 lane config, each pipe needs to have a VC payload size that is a multiple of 2.

In a x4 lane config, each pipe needs to have a VC payload size that is a multiple of 1.

There is one instance of these registers per each pipe timing generator A/B/C/EDP.

Pipe Video Data Island Packet

Data Island Packet (DIP) is a mechanism that allows data to be sent over a digital port during blanking, according to the HDMI and DisplayPort specifications. This includes header, payload, checksum, and ECC information.

Each type of Video DIP will be sent once each frame while it is enabled.

The audio subsystem is also capable of sending Data Island Packets. These packets are programmed by the audio driver and can be read by in MMIO space via the audio control state and audio HDMI widget data island registers.

Video DIP data write sequence:

Wait for 1 VSync to ensure completion of any pending video DIP transmissions.

Disable the video DIP being updated.

Program video DIP data buffer registers for DIP being updated.

Enable the video DIP.

The video DIP data and ECC buffers may be read at any time.

DIP data buffer registers must be programmed with valid data before enabling the DIP.

Partial DIPs are never sent out while the port is enabled. Disabling the DIP at the same time it is being transferred will result in the DIP being completed before the function is disabled.

Shutting off the port on which DIP is being transmitted will result in partial transfer of DIP data. There is no need to switch off the DIP enable bit if the port transmitting DIP is disabled.

When disabling both the DIP port and DIP transmission, first disable the port and then disable DIP.

Enabling a DIP function at the same time that the DIP would have been sent out (had it already been enabled) will result in the DIP being sent on the following frame.

For HDMI, even if no DIP is enabled, a single Null DIP will be sent at the same point in the stream that DIP packets would have been sent.

There is one instance of these registers per each pipe timing generator A/B/C/EDP.

VIDEO_DIP_CTL

VIDEO_DIP_DATA

Construction of DIP for AVI, VS, or SPD (HDMI only):

Dword	Byte3	Byte2	Byte1	Byte0
0	Reserved	HB2	HB1	HB0
1	DB3	DB2	DB1	DB0
2	DB7	DB6	DB5	DB4
3	DB11	DB10	DB9	DB8
4	DB15	DB14	DB13	DB12
5	DB19	DB18	DB17	DB16
6	DB23	DB22	DB21	DB20
7	DB27	DB26	DB25	DB24
8 (RO)	Reserved	Reserved	Reserved	HB ECC
9 (RO)	DB ECC 3	DB ECC 2	DB ECC 1	DB ECC 0

HB = Header Byte

DB = Data Byte

DP = DisplayPort

RO = Read Only

Construction of DIP for GMP (HDMI or DisplayPort):

Dword	Byte3	Byte2	Byte1	Byte0
0	DP: HB3 HDMI: Reserved	HB2	HB1	HB0
1	DB3	DB2	DB1	DB0
2	DB7	DB6	DB5	DB4
3	DB11	DB10	DB9	DB8
4	DB15	DB14	DB13	DB12
5	DB19	DB18	DB17	DB16
6	DB23	DB22	DB21	DB20
7	DB27	DB26	DB25	DB24
8 (RO)	Reserved	Reserved	Reserved	DP: Reserved HDMI: HB ECC
9 (RO)	DP: Reserved HDMI: DB ECC 3	DP: Reserved HDMI: DB ECC 2	DP: Reserved HDMI: DB ECC 1	DP: Reserved HDMI: DB ECC 0
10 (RO)	DP: HB ECC 3 HDMI: Reserved	DP: HB ECC 2 HDMI: Reserved	DP: HB ECC 1 HDMI: Reserved	DP: HB ECC 0 HDMI: Reserved
11 (RO)	DP: DB ECC 3 HDMI: Reserved	DP: DB ECC 2 HDMI: Reserved	DP: DB ECC 1 HDMI: Reserved	DP: DB ECC 0 HDMI: Reserved
12 (RO)	DP: DB ECC 7 HDMI: Reserved	DP: DB ECC 6 HDMI: Reserved	DP: DB ECC 5 HDMI: Reserved	DP: DB ECC 4 HDMI: Reserved

HB = Header Byte

DB = Data Byte

DP = DisplayPort

RO = Read Only

Construction of DIP for VSC (DisplayPort only):

Dword	Byte3	Byte2	Byte1	Byte0
0	HB3	HB2	HB1	HB0
1	DB3	DB2	DB1	DB0
2	DB7	DB6	DB5	DB4
3	DB11	DB10	DB9	DB8
4	DB15	DB14	DB13	DB12
5	DB19	DB18	DB17	DB16
6	DB23	DB22	DB21	DB20
7	DB27	DB26	DB25	DB24
8	DB31	DB30	DB29	DB28
9 (RO)	HB ECC 3	HB ECC 2	HB ECC 1	HB ECC 0
10 (RO)	DB ECC 3	DB ECC 2	DB ECC 1	DB ECC 0
11 (RO)	DB ECC 7	DB ECC 6	DB ECC 5	DB ECC 4

HB = Header Byte

DB = Data Byte

DP = DisplayPort

RO = Read Only

VIDEO_DIP_ECC

Construction of DIP for AVI, VS, or SPD (HDMI only):

Dword	Byte3	Byte2	Byte1	Byte0
0	Reserved	HB2	HB1	HB0
1	DB3	DB2	DB1	DB0
2	DB7	DB6	DB5	DB4
3	DB11	DB10	DB9	DB8
4	DB15	DB14	DB13	DB12
5	DB19	DB18	DB17	DB16
6	DB23	DB22	DB21	DB20
7	DB27	DB26	DB25	DB24
8 (RO)	Reserved	Reserved	Reserved	HB ECC
9 (RO)	DB ECC 3	DB ECC 2	DB ECC 1	DB ECC 0

HB = Header Byte

DB = Data Byte

DP = DisplayPort

RO = Read Only

Construction of DIP for GMP (HDMI or DisplayPort):

Dword	Byte3	Byte2	Byte1	Byte0
0	DP: HB3 HDMI: Reserved	HB2	HB1	HB0
1	DB3	DB2	DB1	DB0
2	DB7	DB6	DB5	DB4
3	DB11	DB10	DB9	DB8
4	DB15	DB14	DB13	DB12
5	DB19	DB18	DB17	DB16
6	DB23	DB22	DB21	DB20
7	DB27	DB26	DB25	DB24
8 (RO)	Reserved	Reserved	Reserved	DP: Reserved HDMI: HB ECC
9 (RO)	DP: Reserved HDMI: DB ECC 3	DP: Reserved HDMI: DB ECC 2	DP: Reserved HDMI: DB ECC 1	DP: Reserved HDMI: DB ECC 0
10 (RO)	DP: HB ECC 3 HDMI: Reserved	DP: HB ECC 2 HDMI: Reserved	DP: HB ECC 1 HDMI: Reserved	DP: HB ECC 0 HDMI: Reserved
11 (RO)	DP: DB ECC 3 HDMI: Reserved	DP: DB ECC 2 HDMI: Reserved	DP: DB ECC 1 HDMI: Reserved	DP: DB ECC 0 HDMI: Reserved
12 (RO)	DP: DB ECC 7 HDMI: Reserved	DP: DB ECC 6 HDMI: Reserved	DP: DB ECC 5 HDMI: Reserved	DP: DB ECC 4 HDMI: Reserved

HB = Header Byte

DB = Data Byte

DP = DisplayPort

RO = Read Only

Construction of DIP for VSC (DisplayPort only):

Dword	Byte3	Byte2	Byte1	Byte0
0	HB3	HB2	HB1	HB0
1	DB3	DB2	DB1	DB0
2	DB7	DB6	DB5	DB4
3	DB11	DB10	DB9	DB8
4	DB15	DB14	DB13	DB12
5	DB19	DB18	DB17	DB16
6	DB23	DB22	DB21	DB20
7	DB27	DB26	DB25	DB24
8	DB31	DB30	DB29	DB28
9 (RO)	HB ECC 3	HB ECC 2	HB ECC 1	HB ECC 0
10 (RO)	DB ECC 3	DB ECC 2	DB ECC 1	DB ECC 0
11 (RO)	DB ECC 7	DB ECC 6	DB ECC 5	DB ECC 4

HB = Header Byte

DB = Data Byte

DP = DisplayPort

RO = Read Only

VIDEO_DIP_GCP

Pipe DDI Function

There is one instance of these registers per each pipe timing generator A/B/C/EDP.

PIPE_DDI_FUNC_CTL

PIPE_MSA_MISC

DDI Buffer

There is one instance of these registers per each DDI A/B/C/D.

DDI_BUF_CTL

DDI Buffer Translation 1 Format

DDI Buffer Translation 2 Format

DDI_BUF_TRANS

Recommended buffer translation programming for DisplayPort							
Entry	**Voltage Swing Level**[1]	**Pre-emphasis Level**[1]	**Non-Transition mV diff p-p**	**Transition mV diff p-p**	**Pre-emphasis dB**	**Dword 1 [31:0]**	**Dword 0 [31:0]**
0	0	0	400	400	0	0006 000Eh	00FF FFFFh
1	0	1	400	600	3.5	0005 000Ah	00D7 5FFFh
2	0	2	400	800	6	0004 0006h	00C3 0FFFh
3	0	3	400	1000	8	000B 0000h	80AA AFFFh
4	1	0	600	600	0	0005 000Ah	00FF FFFFh
5	1	1	600	900	3.5	000C 0004h	00D7 5FFFh
6	1	2	600	1000	4.5	000B 0000h	80C3 0FFFh
7	2	0	800	800	0	0004 0006h	00FF FFFFh
8	2	1	800	1000	2	000B 0000h	80D7 5FFFh
9	Entry 9 is only used for HDMI and DVI. See programming for HDMI and DVI.						

[1]The voltage swing level and pre-emphasis level values follow the naming used in the DisplayPort standard.

Recommended buffer translation programming for FDI							
Entry	**Voltage Swing Level**[1]	**Pre-emphasis Level**[1]	**Non-Transition mV diff p-p**	**Transition mV diff p-p**	**Pre-emphasis dB**	**Dword 1 [31:0]**	**Dword 0 [31:0]**
0	0	0	400	400	0	0007 000Eh	00FF FFFFh
1	0	1	400	600	3.5	000F 000Ah	00D7 5FFFh
2	0	2	400	800	6	0006 0006h	00C3 0FFFh
3	0	3	400	1000	8	001E 0000h	00AA AFFFh
4	1	0	600	600	0	000F 000Ah	00FF FFFFh
5	1	1	600	900	3.5	0016 0004h	00D7 5FFFh
6	1	2	600	1000	4.5	001E 0000h	00C3 0FFFh
7	2	0	800	800	0	0006 0006h	00FF FFFFh
8	2	1	800	1000	2	001E 0000h	00D7 5FFFh
9	Entry 9 is only used for HDMI and DVI. See programming for HDMI and DVI.						

[1]The voltage swing level and pre-emphasis level values follow the naming used in the DisplayPort standard.

Recommended buffer translation programming for HDMI and DVI					
Entry	**Non-Transition mV diff p-p**	**Transition mV diff p-p**	**Pre-emphasis dB**	**Dword 1 [31:0]**	**Dword 0 [31:0]**
0-8	Entries 0 through 8 are only used for DisplayPort and FDI. See programming for DisplayPort and FDI.				
9	800	800	0	0004 0006h	00FF FFFFh
	Alternate values for HDMI and DVI are listed below.				

Alternate values for HDMI and DVI					
Entry	**Non-Transition mV diff p-p**	**Transition mV diff p-p**	**Pre-emphasis dB**	**Dword 1 [31:0]**	**Dword 0 [31:0]**
9	400	400	0	0006 000Eh	00FF FFFFh
9	400	500	2	000E 000Ch	00E7 9FFFh
9	400	600	3.5	0005 000Ah	00D7 5FFFh
9	600	600	0	0005 000Ah	00FF FFFFh
9	600	750	2	001D 0007h	00E7 9FFFh
9	600	900	3.5	000C 0004h	00D7 5FFFh
9	800	800	0	0004 0006h	00FF FFFFh
9	800	1000	2	0003 0002h	80E7 9FFFh
9	850	850	0	0014 0005h	00FF FFFFh
9	900	900	0	000C 0004h	00FF FFFFh
9	950	950	0	001C 0003h	00FF FFFFh
9	1000	1000	0	0003 0002h	80FF FFFFh

DDI AUX Channel

DDI_AUX_CTL

DDI_AUX_DATA

DisplayPort Transport

There is one instance of these registers per each DDI A/B/C/D/E.

DP_TP_CTL

DP_TP_STATUS

SRD

SRD enable sequence:

- Pre-requisite: The associated transcoder and port are running.

1. Configure FBC host and render tracking. The FBC function does not need to be enabled in FBC_CTL.
2. Program Pipe EDP VSC DIP data with a valid setting for SRD/PSR.
3. Enable the VSC in eDP VIDEO_DIP_CTL
4. Configure and enable SRD_CTL

SRD disable sequence:

- Pre-requisite: The associated transcoder and port are running.

1. Disable SRD_CTL.
2. Wait for SRD_STATUS to show SRD is Idle.
3. Wait for vblank
4. Disable VSC

SRD_CTL

SRD_STATUS

SRD Interrupt Bit Definition

SRD_IMR

SRD_IIR

SRD_AUX_CTL

SRD_AUX_DATA

Global Time Code

GTC_CPU_CTL

GTC_CPU_MISC

GTC_CPU_DDA_M

GTC_CPU_DDA_N

GTC_CPU_LIVE

GTC_CPU_REMOTE_CURR

GTC_CPU_LOCAL_CURR

GTC_CPU_REMOTE_PREV

GTC_CPU_LOCAL_PREV

GTC CPU Interrupt Bit Definition

GTC_CPU_IMR

GTC_CPU_IIR

North Display Engine Panel Fitter

There are three panel fitters:

Panel fitter 0 is always 7x5 filter capable.

Panel fitter 1 defaults to 3x3 filter capable. It can be changed to 7x5 filter capable if Panel fitter 2 is disabled and PF1 7x5 Reconfig Enable is selected.

Panel fitter 2 is always 3x3 filter capable.

Any of the three panel fitters can be assigned to any pipe.

A 3x3 capable filter can support pipe horizontal source sizes less than or equal to 2048 pixels. It must not be enabled when the pipe horizontal source size is greater than 2048 pixels.

A 7x5 capable filter can support pipe horizontal source sizes of less than or equal to 4096 pixels. When the pipe horizontal source size is greater than 2048 pixels, the filter will automatically switch to a 3x3 filter mode.

There is one instance of these registers per each panel fitter 0/1/2.

PF_WIN_POS

PF_WIN_SZ

PF_CTRL

North Display Engine I/O Controls

DISPIO_CR_TX_BMU_CR4

North Display Engine Pipe and Plane Controls

Pipe Control

Some of these registers are associated with the timing generator and some with the planes. See the description field of each register.

PIPE_SCANLINE

PIPE_SCANLINECOMP

PIPE_CONF

PIPE_FRMCNT

PIPE_FLIPCNT

PIPE_FRMTMSTMP

PIPE_FLIPTMSTMP

Cursor Plane

CUR_CTL

CUR_BASE

CUR_POS

CUR_PAL

CUR_FBC_CTL

PLANE_SURFLIVE

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

Primary Plane

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

PRI_CTL

PRI_STRIDE

PRI_SURF

PRI_LEFT_SURF

PRI_OFFSET

PLANE_SURFLIVE

Plane Source Pixel Format Mapping of Bits to Colors:

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

Sprite Plane

SPR_CTL

SPR_STRIDE

SPR_POS

SPR_SIZE

SPR_SURF

SPR_LEFT_SURF

SPR_OFFSET

SPR_KEYVAL

SPR_KEYMSK

SPR_KEYMAX

SPR_GAMC REFERENCE POINT FORMAT

SPR_GAMC

SPR_GAMC16

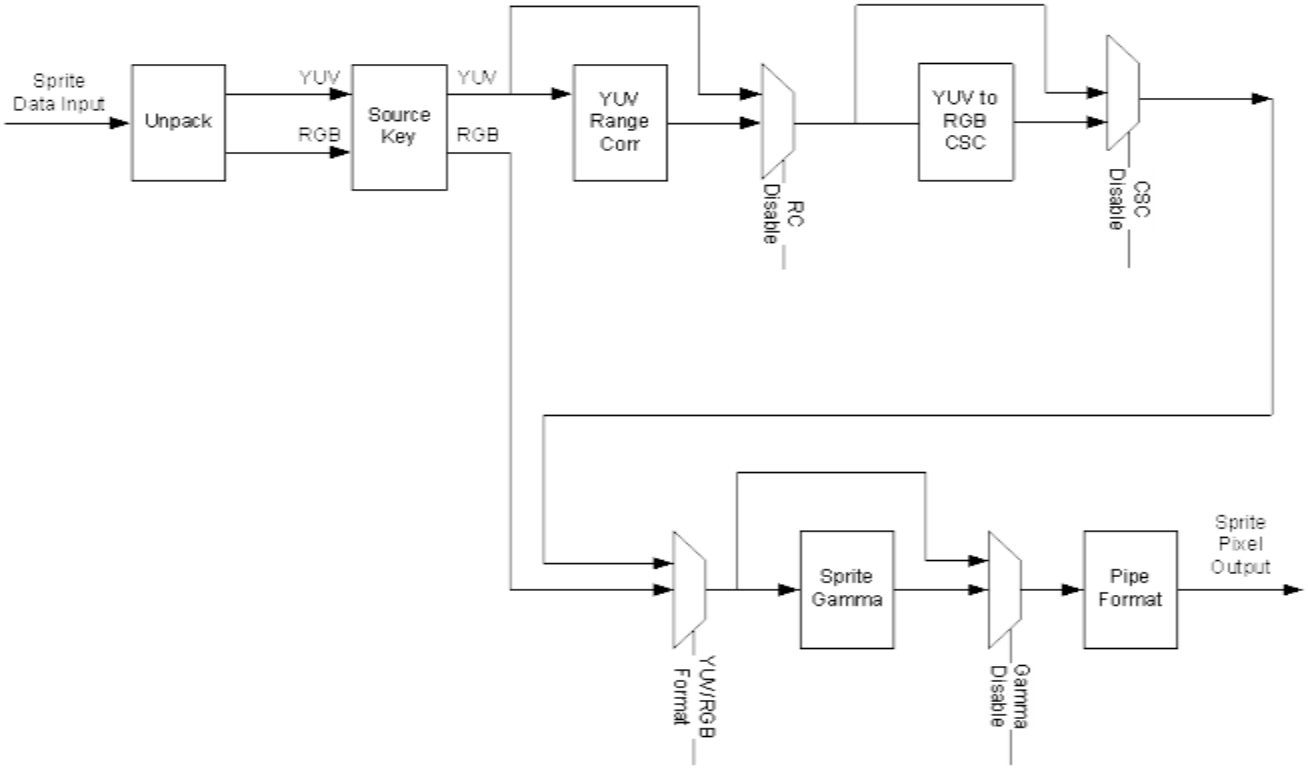
SPR_GAMC17

PLANE_SURFLIVE

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

Data flow through the sprite plane (Steps 2-5 may be enabled or disabled by programming control bits):

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



Sprite Source Pixel Format Mapping of Bits to Colors:

Sprite YUV 4:2:2 Formats	Y1	U	Y2	V
YUV 4:2:2 YUYV	7:0	15:8	23:16	31:24
YUV 4:2:2 UYVY	15:8	7:0	31:24	23:16
YUV 4:2:2 YVYU	7:0	31:24	23:16	15:8
YUV 4:2:2 VYUY	15:8	23:16	31:24	7:0

Sprite YUV 4:4:4 Formats	Ignored	Y	U	V
YUV 32-bit 4:4:4	31:24	23:16	15:8	7:0

Sprite RGB Formats	Ignored	Red	Green	Blue
RGB 32-bit 2:10:10:10 BGRX	31:30	29:20	19:10	9:0
RGB 32-bit 2:10:10:10 RGBX	31:30	9:0	19:10	29:20
RGB 32-bit 2:8:8:8 BGRX	31:24	23:16	15:8	7:0
RGB 32-bit 2:8:8:8 RGBX	31:24	7:0	15:8	23:16
RGB 64-bit 16:16:16:16 BGRX Each component is 1:5:10 MSb-sign:exponent:fraction	63:48	47:32	31:16	15:0
RGB 64-bit 16:16:16:16 RGBX Each component is 1:5:10 MSb-sign:exponent:fraction	63:48	15:0	31:16	47:32
RGB 32-bit XR_BIAS 10:10:10	31:30	9:0	19:10	29:20

North Display Engine Audio

Audio Programming Sequence

The following HDMI and DisplayPort audio programming sequences are for use when enabling or disabling audio or temporarily disabling audio during a display mode set.

The audio codec and audio controller disable sequences must be followed prior to disabling the transcoder or port in a display mode set.

The audio codec and controller enable sequences can be followed after the transcoder is enabled and the port is enabled and completed link training (not sending training or idle patterns if DisplayPort).

The audio controller and audio codec sequences may be done in parallel or serial. In general, the change in ELDV/PD in the codec sequence will generate an unsolicited response to the audio controller driver to indicate that the controller sequence should start, but other mechanisms may be used.

Audio codec disable sequence:

- Disable sample fabrication
    - Set AUD_MISC_CTRL Sample_Fabrication_EN (bit 2) to "0".
- Disable timestamps
    - Set AUD_CONFIG N_value_index (bit 29) to "0" for HDMI or "1" for DisplayPort.
    - Set N_programming_enable (bit 28) to "1"
    - Set Upper_N_value and Lower_N_value (bits 27:20, 15:4) to all "0"s.
- Disable ELDV and ELD buffer
    - Set AUD_PIN_ELD_CP_VLD ELD_valid (bit 0, 4, or 8 based on which port is used) to "0"
- Wait for 2 vertical blanks
- Optional: Disable audio PD (Presence Detect)
    - Software may choose to skip this in order to keep PD enabled during a resolution switch.
    - Set AUD_PIN_ELD_CP_VLD Audio_Output_Enable (bit 2, 6, or 10) to "0".

Audio controller disable sequence:

- Program Stream ID to 0 – Verb ID 706
- Disable audio info frames transmission – Verb ID 732
- Disable Digen – Verb ID 70D
- Program the codec to D3 state if needed.

- Audio driver may stop the audio controller DMA engine at this point if needed, but not required.

Audio codec enable sequence:

- Enable audio Presence Detect
  - Set AUD_PIN_ELD_CP_VLD Audio_Output_Enable (bit 2, 6, or 10) to "1".
- Wait for 1 vertical blank
- Load ELD buffer and Enable ELDV
  - Set AUD_PIN_ELD_CP_VLD ELD_valid (bit 0, 4, or 8 based on which port is used) to "1".
- Enable timestamps
  - Set AUD_CONFIG N_value_index (bit 29) to "0" for HDMI or "1" for DisplayPort.
  - Set N_programming_enable (bit 28) to "0".
  - Program Upper_N_value and Lower_N_value (bits 27:20, 15:4) if a non-default N value is needed.
- Enable sample fabrication if this feature is needed
  - Set AUD_MISC_CTRL Sample_Fabrication_EN (bit 2) to "1".

Audio controller enable sequence:

- Program the codec to D0 state if in D3 state.
- Program Stream ID to non zero – Verb ID 706
- Enable audio info frames transmission – Verb ID 732
- Enable Digen – Verb ID 70D
- If audio controller DMA engine is stopped, audio driver can start the DMA engine at this point.

Audio Configuration

AUD_CONFIG

AUD_MISC_CTRL

AUD_VID_DID

AUD_RID

AUD_M_CTS_ENABLE

Audio Power State Format

AUD_PWRST

AUD_EDID_DATA

AUD_INFOFR

AUD_PIN_PIPE_CONN_ENTRY_LNGTH

AUD_PIPE_CONN_SEL_CTRL

AUD_DIP_ELD_CTRL_ST

South Display Engine Registers LPT+

Lynxpoint South Display Engine

The Lynxpoint-H (LPT:H) South Display Engine supports one CRT pixel path (FDI Receiver, transcoder, and CRT DAC), Hot Plug Detection, GPIO, GMBUS, Panel Power Sequencing, Backlight Modulation, DisplayPort Aux Channel, and GTC.

The Lynxpoint-LP (LPT:LP) South Display Engine supports Hot Plug Detection, GPIO, GMBUS, Panel Power Sequencing, Backlight Modulation, DisplayPort Aux Channel, and GTC.

Terminology

Access Field	Description	Should be implemented as
R/W (Read/Write)	The value written into this register will control hardware and is the same value that will be read.	Write data is stored. Read is from the stored data. Stored value is used to control hardware.
Reserved	Unused register bit. Don't assume a value for these bits. Writes have no effect.	Write data is ignored. Read is zero.
MBZ (Must Be Zero)	Always write a zero to this register.	May be implemented as Reserved or as R/W.
PBC (Preserve Bit Contents)	Software must write the original value back to this bit. This allows new features to be added using these bits.	May be implemented as Reserved or as R/W.
Read Only	The read value is determined by hardware. Writes to this bit have no effect.	Write data is ignored. Read is from a status signal or some other internal source.
Write Only	The value written into this register will control hardware. Reads return zero.	Write data is stored. Read is zero. Stored value is used to control hardware.
R/W Clear (Read/Write Clear)	Sticky status bit. Hardware will set the bit, software can clear it with a write of 1b.	Internal hardware events set a sticky bit. Read is from the sticky bit. A write of 1b clears the sticky bit.
Double Buffered	Write when desired and the written value will take effect at the time of the double buffer update point. Reads will return the written value, which is not necessarily the value being currently used to control hardware. 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. Write data is stored into first stage. Read is from the first stage stored data. First stage stored value is transferred to second stage storage at the double buffer update point. Second stage stored value is used to control hardware. Arm/disarm logic may be used for some registers to control the double buffer update point.
Write/Read Status	The value written into this register will control hardware. The read value is determined by hardware.	Write data is stored. Stored value is used to control hardware. Read is from a status signal or some other internal source.

Shared Functions

Fuses and Straps

SFUSE_STRAP

Raw Clock

RAWCLK_FREQ must be programmed to match the raw clock frequency.

DevLPT, DevWPT

DevLPT:H Raw clock frequency = 125 MHz

DevLPT:LP Raw clock frequency = 24 MHz

DevWPT Raw clock frequency = 24 MHz

RAWCLK_FREQ

Interrupts and Hot Plug

DevLPT, DevWPT

South Display Engine Interrupt Bit Definition

SINTERRUPT

SERR_INT

SHOTPLUG_CTL

SHPD_PULSE_CNT

SHPD_FILTER_CNT

Panel Power and Backlight

DevLPT, DevWPT

PP_STATUS

PP_CONTROL

PP_ON_DELAYS

PP_OFF_DELAYS

PP_DIVISOR

SBLC_PWM_CTL1

SBLC_PWM_CTL2

GPIO

GPIO Pin Usage

These GPIO pins allow the support of simple query and control functions such as DDC and I^2C interface protocols. The GMBUS controller can be used to run the interface protocol, or the GPIO pins can be manually programmed for a "bit banging" interface.

The following tables describe the expected GPIO pin to register mapping. OEMs have the ability to remap these functions onto other pins as long as the hardware limitations are observed. The GPIO pins may also be muxed with other functions such that they are only available when the other function is not being used.

GPIO_CTL

DevLPT:H

Port #	Name	Pin	Pull up/down	Description
5	DDID CTLDATA	DDPD_CTRLDATA	No (Weak pull down on reset)	DDC for HDMI/DVI/DisplayPort port D. Digital port D present strap is set if pin is 1 at rising edge of PCH_PWROK.
	DDID CTLCLK	DDPD_CTRLCLK	No	
4	DDIB CTLDATA	DDPB_CTRLDATA	No (Weak pull down on reset)	DDC for HDMI/DVI/DisplayPort port B. Digital port B present strap is set if pin is 1 at rising edge of PCH_PWROK.
	DDIB CTLCLK	DDPB_CTRLCLK	No	
3	DDIC CTLDATA	DDPC_CTRLDATA	No (Weak pull down on reset)	DDC for HDMI/DVI/DisplayPort port C. Digital port C present strap is set if pin is 1 at rising edge of PCH_PWROK.
	DDIC CTLCLK	DDPC_CTRLCLK	No	
0	DAC DDC Data	CRT_DDC_DATA	No	DDC for Analog monitor (VGA/CRT DAC). This cannot be shared with other DDC or I2C pairs due to legacy monitor requirements.
	DAC DDC Clock	CRT_DDC_CLK	No	

DevLPT:LP, DevWPT

Port #	Name	Pin	Pull up/down	Description
4	DDIB CTLDATA	DDPB_CTRLDATA	No (Weak pull down on reset)	DDC for HDMI/DVI/DisplayPort port B. Digital port B present strap is set if pin is 1 at rising edge of PCH_PWROK.
	DDIB CTLCLK	DDPB_CTRLCLK	No	
3	DDIC CTLDATA	DDPC_CTRLDATA	No (Weak pull down on reset)	DDC for HDMI/DVI/DisplayPort port C. Digital port C present strap is set if pin is 1 at rising edge of PCH_PWROK.
	DDIC CTLCLK	DDPC_CTRLCLK	No	

GMBUS

The GMBUS (Graphic Management Bus) is used to access/control devices connected to the GPIO pins.

Basic features:

1. I^2C compatible.
2. Bus clock frequency of 50KHz and 100KHz.
3. Attaches to any of the GPIO pin pairs.
4. 7-bit or 10-bit Slave Address and 8-bit or 16-bit index.
5. Double buffered data register and a 9 bit counter support 0 byte to 256 byte transfers.
6. Supports stalls generated by the slave device pulling down the clock line (Slave Stall), or delaying the slave acknowledge response.
7. Status register indicates error conditions, data buffer busy, time out, and data complete acknowledgement.
8. Detects and reports time out conditions for a stall from a slave device, or a delayed or missing slave acknowledge.
9. Interrupts may optionally be generated.
10. Does not directly support segment pointer addressing as defined by the Enhanced Display Data Channel standard.

Segment pointer addressing as defined by the Enhanced Display Data Channel standard:

1. Use bit bashing (manual GPIO programming) to complete segment pointer write **without terminating in a stop or wait cycle**.
2. Terminate bit bashing phase with both I^2C lines pulled high by tri-stating the data line before the clock line. Follow EDDC requirement for response received from slave device.
3. Initiate GMBUS cycle as required to transfer EDID following normal procedure.

DevLPT, DevWPT

GMBUS0

GMBUS1

GMBUS2

GMBUS3

GMBUS4

GMBUS5

Side Band Interface

The south display pixel clock and the clock references used by south and north display are driven by the PCH internal clock controller (iCLK).

Access to iCLK or FDI mPHY registers must be done through the Sideband Interface (SBI).

See the Display iCLK Programming specification for the values to program through SBI to control the pixel clock, references, and FDI mPHY.

SBI_ADDR

SBI_DATA

SBI_CTL_STAT

SBI write sequence

1. Write address values into address register SBI_ADDR.
2. Write data value into data register SBI_DATA.
3. Write control values and initiate transaction through control and status register SBI_CTL_STAT.
4. Poll on status showing transaction completed in SBI_CTL_STAT.

SBI read sequence

1. Write address values into address register SBI_ADDR.
2. Write control values and initiate transaction through control and status register SBI_CTL_STAT.
3. Poll on status showing transaction completed in SBI_CTL_STAT.
4. Read data result value from data register SBI_DATA.

DisplayPort AUX Channel

DP_AUX_CTL

DP_AUX_DATA

Global Time Code (GTC)

On LPT:LP and WPT, the GTC clock must be enabled prior to enabling GTC, then wait for 40us for warmup, then GTC can be enabled.

GTCCLK_EN

GTC_CTL

GTC_MISC

GTC_DDA_M

GTC_DDA_N

GTC PCH Interrupt Bit Definition

GTC_PCH_IMR

GTC_PCH_IIR

GTC_PORT_CTL

GTC_PORT_RX_CURR

GTC_PORT_TX_CURR

Pixel Path

Pixel Clock

The south display pixel clock is used by the transcoder and analog CRT port.

South Display Modes	South Display Pixel Clock Frequency
Analog CRT Port	25-180 MHz

The pixel clock frequency and enabling is controlled through the PCH internal clock controller (iCLK) which is programmed through the Sideband Interface (SBI). See the Display iCLK Programming specifiction for the values to program through SBI to control the pixel clock.

The display pixel clock must be gated prior to disabling the pixel clock, and kept gated until after the pixel clock is enabled and the warmup period has passed.

PIXCLK_GATE

FDI Receiver

Control

FDI_RX_CTL

FDI_RX_MISC

FDI_RX_TUSIZE

Interrupt

FDI Receiver Interrupt Bit Definition

FDI_RX_IMR

FDI_RX_IIR

Transcoder

Control

TRANS_CONF

Timings

HTOTAL

HBLANK

HSYNC

VTOTAL

VBLANK

VSYNC

VSYNCSHIFT

Analog Port

DAC_CTL

Display iCLK Programming

Display software controls the three integrated clock (iCLK or ICC) resources allocated for display.

All three resources are available on LPT:H non-ULT systems. Only one of the resources is available on LPT:LP ULT systems.

	CLKOUT_DP iCLK Clock 1	CLKOUT_DPNS (LPT:H non-ULT only) iCLK Clock 4	VGA Pixel Clock (LPT:H non-ULT only) iCLK Clock 5
Usage	Spread spectrum reference for CPU display engine PLLs and FDI receiver. Can also be used for clock bending; which uses the spread modulator to give a fine adjustment to the frequency.	Non-spread spectrum reference for CPU display engine PLLs. Only present on LPT:H non-ULT systems.	Pixel clock for the CRT DAC output. Only present on LPT:H non-ULT systems.
Frequency	Programmable selection between 135 MHz with 0.5% downspread, 135 MHz with 0% spread, and 135Mhz bent.	135 MHz	Programmable selection ranging from 20 to 180 MHz.
Source	24 MHz crystal (typical <93 ppm error)	24 MHz crystal (typical <93 ppm error)	24 MHz crystal (typical <93 ppm error)
Default after reset	Disabled	Enabled	Disabled
Programming	Should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence and Programming of CLKOUT_DP.	Should be disabled by BIOS when internal graphics is disabled; as documented in Programming of CLKOUT_DPNS.	Should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence and Programming of VGA Pixel Clock.

Register Definitions

These registers are accessed using the South Display Engine Side Band Interface (SBI). Program the SBI_ADDR offset field with the concatenation of the Target ID (most significant 8 bits) and the Register Start (least significant 8 bits).

SSCDITHPHASE

Target ID			Clock 1 = 0x02
Register Start			0x04
Bit	**Access**	**Default Value**	**Description**
31:1	R/W	00000000h	**cfg_ssc_partialphasepatt<31:1>:** This field is used to set the fractional step for clock bending. The granularity is one 5.787ps adjustment per 32 clock1 periods (nominally clock1 is 135Mhz.) This register field is double-buffered and is propagated under the control of register SSCDIVINTPHASE, bit 0, of same target ID.
0	R/W	0b	**cfg_ssc_partialphasepatt<0>:** 1: Dithering pattern is enabled. 0: Dithering pattern is disabled.

SSCDIVINTPHASE

Target ID			Clock 1 = 0x02 Clock 5 = 0x06
Register Start			0x00
Bit	**Access**	**Default Value**	**Description**
31:15	Reserved	Reserved	Reserved content: Values must not be changed.
14:8	R/W	00h	**ssc_ocphaseincval:** 000_0000: 0 PI phase change 000_0001: 1 PI phase change 000_0010: 2 PI phase change ... 111_1111: 127 PI phase change The granularity of this field is 5.787ps of period adjustment for each step of Integer Phase Value. This register field is double-buffered and is propagated under the control of register SSCDIVINTPHASE, bit 0, of same target ID.
7:1	R/W	12h	**idiv7bcnt_divsel_sscdiv:** N+2 encoding 00h = Reserved 01h = Reserved 02h = Reserved 03h = divide by 5 (MIN) 12h = Div by 20 **Default for 135Mhz clock from 2700Mhz PI clock** 30h = divide by 50 60h = divide by 98 7Fh = divide by 129 (MAX) This register field is double-buffered and is propagated under the control of register SSCDIVINTPHASE, bit 0, of same target ID.

Target ID			Clock 1 = 0x02 Clock 5 = 0x06
Register Start			0x00
Bit	Access	Default Value	Description
0	R/W	0b	**ssc_ocparam_propagate:** Software sets this register to trigger the propagation of register programming and hardware clears this bit when done with the propagation. 0: No pending propagation 1: Pending propagation This field affects registers SSCDIVINTPHASE and SSCDITHPHASE of same target ID.

SSCAUXDIV

Target ID			Clock 5 = 0x06
Register Start			0x10
Bit	**Access**	**Default Value**	**Description**
31:5	Reserved	Reserved	Reserved content: Values must not be changed.
4	R/W	0b	**finaldiv2sel:** Divide final output of the 7-bit divider by 2
3:0	Reserved	Reserved	Reserved content: Values must not be changed.

SSCCTL

Target ID			Clock 1 = 0x02 Clock 4 = 0x05 Clock 5 = 0x06
Register Start			0x0C
Bit	**Access**	**Default Value**	**Description**
31:9	Reserved	Reserved	Reserved content: Values must not be changed.
8	R/W	0b	**ssc_divonlymode:** Power down the SSC
7:4	Reserved	Reserved	Reserved content: Values must not be changed.
3	R/W	0b	**cfg_ssc_patthalt:** Halt the SSC phase pattern to allow on-the-fly spread parameters change.
2:1	Reserved	Reserved	Reserved content: Values must not be changed.
0	R/W	Clock 1 = 1b Clock 4 = 0b Clock 5 = 1b	**cfg_ssc_enable_b:** 0: Enable 1: Disable

DBUFF0/GEN0

Target ID			CLKOUT_DP for LPT:H = 0x2A CLKOUT_DP for LPT:LP = 0x1F CLKOUT_DPNS = 0x2B
Register Start			0x00
Bit	**Access**	**Default Value**	**Description**
31:1	Reserved	Reserved	Reserved content: Values must not be changed.
0	R/W	CLKOUT_DP = 0b CLLOUT_DPNS = 1b	**gen0_cfg/buffenable:** 0: Enable 1: Disable

Programming of CLKOUT_DP

CLKOUT_DP defaults to disabled.

CLKOUT_DP should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence.

Note that the CLKOUT_DP can be used for SSC and FDI, or for clock bending. When bending clock on CLKOUT_DP, FDI cannot be used, and vice-versa.

Sequence to enable CLKOUT_DP

This enables CLKOUT_DP with 0.5% downspread.

Clock bending must be shut off when using spread..

1. Write Clock 1 SSCCTL cfg_ssc_enable_b to '0' and cfg_ssc_patthalt to '1'. This turns on the modulator and associated divider at the source, and disables the spread pattern.
2. Wait for 24us initialization time of the modulator and divider.
3. Write Clock 1 SSCCTL cfg_ssc_patthalt to '0'. This allows the spread pattern to run.
    - Workaround requirement to only enable the spread pattern after the modulator and divider are initialized.
4. Write CLKOUT_DP DBUFF0/GEN0 gen0_cfg/buffenable to '1'. This enables the off-chip output clock to toggle.

Sequence to enable CLKOUT_DP without spread

This enables CLKOUT_DP with no spread.

1. Write Clock 1 SSCCTL cfg_ssc_enable_b to '0' and cfg_ssc_patthalt to '1'. This turns on the modulator and associated divider at the source, and disables the spread pattern.
2. Wait for 24us initialization time of the modulator and divider.
3. Write CLKOUT_DP DBUFF0/GEN0 gen0_cfg/buffenable to '1'. This enables the off-chip output clock to toggle.

Sequence to disable CLKOUT_DP

1. Write CLKOUT_DP DBUFF0/GEN0 gen0_cfg/buffenable to '0'. This gates the off-chip output clock from toggling.
2. If spread is enabled, write Clock 1 SSCCTL cfg_ssc_patthalt to '1'. This halts the spread pattern.
3. If spread was enabled, wait for 32us for spread pattern to return to neutral state.
    - Workaround requirement to disable the spread pattern and allow it to return to the neutral state before the modulator and divider are disabled.
4. Write Clock 1 SSCCTL cfg_ssc_enable_b to '1'. This powers down the modulator and associated divider at the source.

Sequence to bend CLKOUT_DP

Spread must be halted when using bending.

Refer to the table below for the bending step values to program.

When done with bending, program 0.0 steps to shut off bending before using CLKOUT_DP for anything else.

1. Optionally follow Sequence to disable CLKOUT_DP. Once CLKOUT_DP is enabled without spread, bending values can be changed on the fly without disabling and enabling the clock.
2. Write Clock 1 SSCDITHPHASE to set up a half-step adjustment.
    - A half-step amounts to 0.5 * 5.787ps of period adjustment to the baseline 135Mhz clock.
3. Write Clock 1 SSCDIVINTPHASE to set up quantity of full-step adjustment, and to trigger application of changes to SSCDITHPHASE and SSCDIVINTPHASE.
    - A full-step amounts to 5.787ps of period adjustment to the baseline 135Mhz clock. More than one full-step can be specified.
4. Optionally follow Sequence to enable CLKOUT_DP without spread.

Steps	Direction	SSCDITHPHASE	SSCDIVINTPHASE	Nominal Period (ps)	Total Period Adjustment (ps)	Effective Period (ps)	Effective Frequency (Mhz)
5.0	Speed Up	0000_0000h	0000_3B23h	7037.037	341.44	7378.47	135.529
4.5	Speed Up	AAAA_AAABh	0000_3B23h	7037.037	344.33	7381.37	135.476
4.0	Speed Up	0000_0000h	0000_3C23h	7037.037	347.22	7384.26	135.423
3.5	Speed Up	AAAA_AAABh	0000_3C23h	7037.037	350.12	7387.15	135.370
3.0	Speed Up	0000_0000h	0000_3D23h	7037.037	353.01	7390.05	135.317
2.5	Speed Up	AAAA_AAABh	0000_3D23h	7037.037	355.90	7392.94	135.264
2.0	Speed Up	0000_0000h	0000_3E23h	7037.037	358.80	7395.83	135.211
1.5	Speed Up	AAAA_AAABh	0000_3E23h	7037.037	361.69	7398.73	135.158
1.0	Speed Up	0000_0000h	0000_3F23h	7037.037	364.58	7401.62	135.106
0.5	Speed Up	AAAA_AAABh	0000_3F23h	7037.037	367.48	7404.51	135.053
0.0	N/A	0000_0000h	0000_0025h[1]	7407.41	0.00	7407.41	135.000
0.5	Slow Down	AAAA_AAABh	0000_0025h	7407.41	2.89	7410.30	134.947
1.0	Slow Down	0000_0000h	0000_0125h	7407.41	5.79	7413.19	134.895
1.5	Slow Down	AAAA_AAABh	0000_0125h	7407.41	8.68	7416.09	134.842
2.0	Slow Down	0000_0000h	0000_0225h	7407.41	11.57	7418.98	134.789
2.5	Slow Down	AAAA_AAABh	0000_0225h	7407.41	14.47	7421.88	134.737
3.0	Slow Down	0000_0000h	0000_0325h	7407.41	17.36	7424.77	134.684

Steps	Direction	SSCDITHPHASE	SSCDIVINTPHASE	Nominal Period (ps)	Total Period Adjustment (ps)	Effective Period (ps)	Effective Frequency (Mhz)
3.5	Slow Down	AAAA_AAABh	0000_0325h	7407.41	20.25	7427.66	134.632
4.0	Slow Down	0000_0000h	0000_0425h	7407.41	23.15	7430.56	134.579
4.5	Slow Down	AAAA_AAABh	0000_0425h	7407.41	26.04	7433.45	134.527
5.0	Slow Down	0000_0000h	0000_0525h	7407.41	28.94	7436.34	134.475

[1]Hardware default has bending off. However, once bending has been enabled, to shut it off requires register writes of these values for 0.0 steps.

Programming of CLKOUT_DPNS

CLKOUT_DPNS exists only on LPT:H Non-ULT systems.

CLKOUT_DPNS defaults to enabled with 135Mhz non-spread, non-bent.

CLKOUT_DPNS should be disabled by BIOS when internal graphics is disabled and will not be re-enabled.

Sequence to disable CLKOUT_DPNS

1. Write CLKOUT_DPNS DBUFF0/GEN0 gen0_cfg/buffenable to '0'. This gates the off-chip output clock from toggling.
2. Write Clock 4 SSCCTL cfg_ssc_enable_b to '1' and ssc_divonlymode to '0'. This powers down the modulator and associated divider at the source.

Programming of VGA Pixel Clock

VGA Pixel Clock exists only on LPT:H Non-ULT systems.

VGA Pixel Clock defaults to disabled.

VGA Pixel Clock should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence.

Sequence to disable VGA pixel clock

1. Write PIXCLK_GATE Pixel Clock Ungate to '0' to gate the clock.
2. Write Clock 5 SSCCTL cfg_ssc_enable_b to '1'. This powers down the modulator and associated divider at the source.

Sequence to enable VGA pixel clock

Refer to the frequency calculation or frequency look-up table for the values to program.

1. Write Clock 5 SSCDIVINTPHASE idiv7bcnt_divsel_sscdiv and ssc_ocphaseincval with the calculated values.
2. Write Clock 5 SSCAUXDIV finaldiv2sel with the calculated value.
3. Write Clock 5 SSCCTL cfg_ssc_enable_b to '0'. This turns on the modulator and associated divider at the source.
4. Wait for 24us initialization time of the modulator and divider.
5. Write PIXCLK_GATE Pixel Clock Ungate to '1' to ungate the clock.

Frequency Calculation

This procedure results in VGA frequency output of less than 600ppm total error (accounting for crystal input ppm error.)

```
iCLK_virtual_root_freq = 172800; # in Mhz
iCLK_pi_range = 64;
desired_divisor = ROUND (iCLK_virtual_root_freq / desired_VGA_frequency); # round to nearest integer
MSB_divisor_value = INT (desired_divisor / iCLK_pi_range); # MSB value, bias on low divisor value side
PI_value = MOD (desired_divisor / iCLK_pi_range); # PI value, bias on slow down direction
If (desired_VGA_frequency == 20) { # Corner case of 20Mhz
        idiv7bcnt_divsel_sscdiv = "100_0001b";
        ssc_ocphaseincval = "010_0000b";
        finaldiv2sel = "1b";
}
Else { # Other frequencies
        idiv7bcnt_divsel_sscdiv = DEC2BIN (MSB_divisor_value - 2);
        ssc_ocphaseincval = DEC2BIN(PI_value);
        finaldiv2sel = "0b";
 };
```

Frequency Look-Up Table

finaldiv2sel SSCAUXDIV[4]	idiv7bcnt_divsel_sscdiv SSCDIVINTPHASE[7:1] (Hex)	ssc_ocphaseincval SSCDIVINTPHASE[14:8] (Hex)	Target Frequency (MHz)	Effective Frequency (Mhz)	PPM Error[1]
1	41	20	20.000	20.000	0
0	7E	25	21.000	20.999	-53
0	79	0E	21.912	21.912	12
0	78	2F	22.000	21.999	-58
0	73	19	23.000	23.000	6
0	6E	20	24.000	24.000	0
0	6A	00	25.000	25.000	0
0	69	10	25.175	25.175	-7
0	69	09	25.200	25.201	21
0	65	36	26.000	26.001	24
0	62	00	27.000	27.000	0
0	61	3A	27.027	27.025	-62
0	60	0C	27.500	27.498	-58
0	5E	1B	28.000	28.002	70
0	5D	16	28.320	28.319	-50
0	5D	15	28.322	28.323	44
0	5B	07	29.000	28.998	-64
0	58	00	30.000	30.000	0
0	55	06	31.000	31.001	35
0	53	2E	31.500	31.498	-53
0	52	18	32.000	32.000	0
0	51	05	32.500	32.500	-15
0	4F	34	33.000	33.002	70
0	4D	1A	34.000	34.002	70
0	4B	09	35.000	35.001	29
0	4A	04	35.500	35.497	-82
0	49	00	36.000	36.000	0
0	46	3E	37.000	37.002	58
0	45	03	38.000	38.003	82
0	43	0F	39.000	38.998	-53
0	41	20	40.000	40.000	0
0	40	2B	40.500	40.497	-79

finaldiv2sel SSCAUXDIV[4]	idiv7bcnt_divsel_sscdiv SSCDIVINTPHASE[7:1] (Hex)	ssc_ocphaseincval SSCDIVINTPHASE[14:8] (Hex)	Target Frequency (MHz)	Effective Frequency (Mhz)	PPM Error[1]
0	40	26	40.541	40.544	95
0	3F	37	41.000	40.996	-87
0	3F	00	41.540	41.538	-38
0	3E	12	42.000	42.003	70
0	3C	33	43.000	42.996	-99
0	3C	23	43.163	43.168	108
0	3B	17	44.000	44.003	70
0	3A	09	44.900	44.895	-117
0	3A	00	45.000	45.000	0
0	38	2D	46.000	45.994	-128
0	37	1D	47.000	46.995	-110
0	36	10	48.000	48.000	0
0	35	07	49.000	48.993	-134
0	34	23	49.500	49.499	-27
0	34	00	50.000	50.000	0
0	32	3C	51.000	51.004	70
0	31	3B	52.000	52.001	24
0	31	21	52.406	52.411	101
0	30	3C	53.000	53.006	116
0	30	00	54.000	54.000	0
0	2F	3D	54.054	54.051	-62
0	2F	06	55.000	54.997	-58
0	2E	0E	56.000	55.995	-93
0	2E	00	56.250	56.250	0
0	2D	18	57.000	56.992	-139
0	2C	23	58.000	58.006	105
0	2B	31	59.000	58.996	-64
0	2B	00	60.000	60.000	0
0	2A	11	61.000	60.995	-76
0	29	23	62.000	62.002	35
0	28	37	63.000	62.997	-53
0	28	0C	64.000	64.000	0
0	27	22	65.000	65.011	174
0	26	3A	66.000	66.005	70

finaldiv2sel SSCAUXDIV[4]	idiv7bcnt_divsel_sscdiv SSCDIVINTPHASE[7:1] (Hex)	ssc_ocphaseincval SSCDIVINTPHASE[14:8] (Hex)	Target Frequency (MHz)	Effective Frequency (Mhz)	PPM Error[1]
0	26	20	66.667	66.667	-5
0	26	13	67.000	67.003	41
0	25	2D	68.000	68.005	70
0	25	27	68.179	68.166	-196
0	25	08	69.000	69.010	139
0	24	25	70.000	69.988	-174
0	24	02	71.000	70.994	-82
0	23	20	72.000	72.000	0
0	22	3F	73.000	73.004	53
0	22	1F	74.000	74.004	58
0	22	1A	74.175	74.163	-161
0	22	17	74.250	74.259	118
0	22	10	74.481	74.483	24
0	22	00	75.000	75.000	0
0	21	22	76.000	75.989	-139
0	21	04	77.000	77.005	70
0	20	27	78.000	78.014	174
0	20	12	78.750	78.760	131
0	20	0B	79.000	79.012	157
0	1F	30	80.000	80.000	0
0	1F	15	81.000	81.013	157
0	1F	13	81.081	81.089	95
0	1F	05	81.624	81.625	12
0	1E	3B	82.000	82.012	151
0	1E	22	83.000	82.997	-35
0	1E	0A	83.950	83.965	179
0	1E	09	84.000	84.006	70
0	1D	31	85.000	84.998	-29
0	1D	19	86.000	86.013	151
0	1D	02	87.000	87.009	105
0	1C	2C	88.000	87.984	-186
0	1C	16	89.000	88.980	-220
0	1C	00	90.000	90.000	0
0	1B	2B	91.000	90.995	-53

finaldiv2sel SSCAUXDIV[4]	idiv7bcnt_divsel_sscdiv SSCDIVINTPHASE[7:1] (Hex)	ssc_ocphaseincval SSCDIVINTPHASE[14:8] (Hex)	Target Frequency (MHz)	Effective Frequency (Mhz)	PPM Error[1]
0	1B	16	92.000	92.013	139
0	1B	02	93.000	93.003	35
0	1A	2E	94.000	94.015	163
0	1A	25	94.500	94.478	-235
0	1A	1B	95.000	94.997	-29
0	1A	0F	95.654	95.628	-271
0	1A	08	96.000	96.000	0
0	19	35	97.000	97.024	249
0	19	23	98.000	98.015	151
0	19	11	99.000	99.026	261
0	19	00	100.000	100.000	0
0	18	2F	101.000	100.994	-64
0	18	1E	102.000	102.007	70
0	18	0E	103.000	102.980	-197
0	17	3E	104.000	103.971	-278
0	17	2E	105.000	104.982	-174
0	17	1E	106.000	106.012	116
0	17	0F	107.000	106.997	-29
0	17	0C	107.214	107.196	-168
0	17	00	108.000	108.000	0
0	16	31	109.000	109.022	203
0	16	23	110.000	109.994	-58
0	16	23	110.013	109.994	-177
0	16	15	111.000	110.983	-157
0	16	11	111.263	111.269	55
0	16	10	111.375	111.340	-313
0	16	07	112.000	111.990	-93
0	15	39	113.000	113.015	134
0	15	35	113.309	113.311	22
0	15	38	113.100	113.089	-98
0	15	2C	114.000	113.984	-139
0	15	1F	115.000	114.970	-261
0	15	12	116.000	115.973	-232
0	15	05	117.000	116.994	-53

finaldiv2sel SSCAUXDIV[4]	idiv7bcnt_divsel_sscdiv SSCDIVINTPHASE[7:1] (Hex)	ssc_ocphaseincval SSCDIVINTPHASE[14:8] (Hex)	Target Frequency (MHz)	Effective Frequency (Mhz)	PPM Error[1]
0	14	38	118.000	118.033	278
0	14	2C	119.000	119.008	70
0	14	24	119.651	119.668	139
0	14	20	120.000	120.000	0
0	14	14	121.000	121.008	70
0	14	08	122.000	122.034	278
0	14	01	122.614	122.640	214
0	13	3D	123.000	122.989	-87
0	13	39	123.379	123.340	-313
0	13	32	124.000	123.960	-324
0	13	26	125.000	125.036	290
0	13	1B	126.000	126.039	313
0	13	11	127.000	126.965	-272
0	13	06	128.000	128.000	0
0	12	3C	129.000	128.955	-348
0	12	33	129.859	129.827	-245
0	12	31	130.000	130.023	174
0	12	27	131.000	131.008	64
0	12	1F	131.850	131.808	-321
0	12	1D	132.000	132.009	70
0	12	13	133.000	133.025	192
0	12	10	133.330	133.333	26
0	12	0A	134.000	133.953	-348
0	12	00	135.000	135.000	0
0	11	37	136.000	135.956	-324
0	11	2D	137.000	137.034	249
0	11	24	138.000	138.019	139
0	11	1B	139.000	139.019	134
0	11	1B	139.050	139.019	-227
0	11	1B	139.054	139.019	-256
0	11	12	140.000	140.032	232
0	11	0A	141.000	140.946	-382
0	11	01	142.000	141.988	-82
0	10	38	143.000	143.046	325

finaldiv2sel SSCAUXDIV[4]	idiv7bcnt_divsel_sscdiv SSCDIVINTPHASE[7:1] (Hex)	ssc_ocphaseincval SSCDIVINTPHASE[14:8] (Hex)	Target Frequency (MHz)	Effective Frequency (Mhz)	PPM Error[1]
0	10	34	143.472	143.522	346
0	10	30	144.000	144.000	0
0	10	28	145.000	144.966	-232
0	10	20	146.000	145.946	-371
0	10	18	147.000	146.939	-417
0	10	10	147.891	147.945	367
0	10	10	148.000	147.945	-371
0	10	0D	148.350	148.326	-161
0	10	0C	148.500	148.454	-313
0	10	08	149.000	148.966	-232
0	10	00	150.000	150.000	0
0	0F	38	151.000	151.049	325
0	0F	31	152.000	151.979	-139
0	0F	2F	152.280	152.247	-219
0	0F	29	153.000	153.056	365
0	0F	22	154.000	154.011	70
0	0F	1B	155.000	154.978	-145
0	0F	14	156.000	155.957	-278
0	0F	0D	157.000	156.948	-330
0	0F	09	157.500	157.521	131
0	0F	06	158.000	157.952	-301
0	0E	3F	159.000	158.970	-191
0	0E	38	160.000	160.000	0
0	0E	31	161.000	161.044	273
0	0E	2B	162.000	161.949	-313
0	0E	24	163.000	163.019	116
0	0E	1E	164.000	163.947	-324
0	0E	17	165.000	165.043	261
0	0E	11	166.000	165.994	-35
0	0E	0B	167.000	166.957	-261
0	0E	05	168.000	167.930	-417
0	0D	3E	169.000	169.080	475
0	0D	3E	169.128	169.080	-283
0	0D	38	170.000	170.079	464

finaldiv2sel SSCAUXDIV[4]	idiv7bcnt_divsel_sscdiv SSCDIVINTPHASE[7:1] (Hex)	ssc_ocphaseincval SSCDIVINTPHASE[14:8] (Hex)	Target Frequency (MHz)	Effective Frequency (Mhz)	PPM Error[1]
0	0D	33	171.000	170.920	-469
0	0D	2D	172.000	171.940	-348
0	0D	28	172.800	172.800	0
0	0D	27	173.000	172.973	-157
0	0D	21	174.000	174.018	105
0	0D	1D	174.787	174.722	-373
0	0D	1B	175.000	175.076	435
0	0D	19	175.500	175.431	-391
0	0D	16	176.000	175.967	-186
0	0D	10	177.000	177.049	278
0	0D	0B	178.000	177.961	-220
0	0D	05	179.000	179.067	377
0	0D	00	180.000	180.000	0

[1] PPM error budget is 1000. Total ppm error is the number specified in this table plus the ppm error of the 25Mhz external crystal. The crystal ppm error is typically 93 (accounting for crystal cut accuracy, temperature variation, aging, loading variation, etc.)

Programming of FDI Reference and mPHY

FDI exists only on LPT:H Non-ULT systems.

FDI uses the CLKOUT_DP reference with 0.5% downspread.

CLKOUT_DP defaults to disabled.

CLKOUT_DP should be enabled and disabled by software as needed when enabling and disabling a display; as documented in Display Mode Set Sequence.

PCH FDI I/O (FDI mPHY) must be configured as part of CLKOUT_DP enabling when FDI will be used.

On LPT:H PCH FDI I/O must be configured at boot, even if FDI will not be used.

CLKOUT_DP must be running before PCH FDI I/O is configured. FDI mPHY must be reset between when CLKOUT_DP is enabled for FDI and FDI mPHY registers are programmed.

Note that the CLKOUT_DP can be used for SSC and FDI, or for clock bending. When bending clock on CLKOUT_DP, FDI cannot be used, and vice-versa.

Sequence to enable CLKOUT_DP for FDI usage and configure PCH FDI I/O

1. Write Clock 1 SSCCTL cfg_ssc_enable_b to '0' and cfg_ssc_patthalt to '1'. This turns on the modulator and associated divider at the source, and disables the spread pattern.
2. Wait for 24us initialization time of the modulator and divider.
3. Write Clock 1 SSCCTL cfg_ssc_patthalt to '0'. This allows the spread pattern to run.
    - Workaround requirement to only enable the spread pattern after the modulator and divider are initialized.
4. Reset FDI mPHY
    a. Write register 0xC2004 bit 12 = 1b to reset FDI mPHY.
    b. Poll for register 0xC2004 bit 13 = 1b to indicate reset asserted. Timeout and fail after 100 µS.
    c. Write register 0xC2004 bit 12 = 0b to clear reset FDI mPHY.
    d. Poll for register 0xC2004 bit 13 = 0b to indicate reset de-asserted. Timeout and fail after 100 µS.
5. Program FDI mPHY registers through Side Band Interface, using the values listed in the FDI mPHY Register Programming table below. Use read/modify/write to avoid changing other bits in those registers.
6. Write CLKOUT_DP DBUFF0/GEN0 gen0_cfg/buffenable to '1'. This enables the off-chip output clock to toggle.

Sequence to disable CLKOUT_DP for FDI usage

- Follow Programming of CLKOUT_DP, Sequence to disable CLKOUT_DP.

Sequence to configure PCH FDI I/O

Follow steps 1 through 5 of Sequence to enable CLKOUT_DP for FDI usage and configure PCH FDI I/O

6. Write Clock 1 SSCCTL cfg_ssc_patthalt to '1'. This halts the spread pattern.
7. Wait for 32us for spread pattern to return to neutral state.
    - Workaround requirement to disable the spread pattern and allow it to return to the neutral state before the modulator and divider are disabled.
8. Write Clock 1 SSCCTL cfg_ssc_enable_b to '1'. This powers down the modulator and associated divider at the source.

FDI mPHY Register Programming

These registers are accessed using the South Display Engine Side Band Interface (SBI). Program the SBI_ADDR offset field with the offsets from the table.

Offset	Field	Bits	Value
0x8008	idtclkmdiv[7:0]	[31:24]	0001_0010b
0x2008	lane0 cri_rxrawdata_sel	[11]	1b
0x2108	lane1 cri_rxrawdata_sel	[11]	1b
0x206C	lane0 ofrcgohighzen	[24]	1b
0x206C	lane0 ofrctxclkrootdis	[21]	1b
0x206C	lane0 ofrcpwrmodel1	[18]	1b
0x216C	lane1 ofrcgohighzen	[24]	1b
0x216C	lane1 ofrctxclkrootdis	[21]	1b
0x216C	lane1 ofrcpwrmodel1	[18]	1b
0x2080	lane0 ocfgpiobuf[2:0]	[15:13]	101b
0x2180	lane1 ocfgpiobuf[2:0]	[15:13]	101b
0x208C	lane0 icfgpfcfg_gen2[7:0]	[7:0]	0001_1100b
0x218C	lane1 icfgpfcfg_gen2[7:0]	[7:0]	0001_1100b
0x2098	lane0 icfgpfcfg_gen1[7:0]	[23:16]	0001_1100b
0x2198	lane1 icfgpfcfg_gen1[7:0]	[23:16]	0001_1100b
0x20C4	lane0 piobufcfg_override	[27]	1b
0x21C4	lane1 piobufcfg_override	[27]	1b
0x20EC	lane0 cal_num[3:0]	[31:28]	0100b
0x21EC	lane1 cal_num[3:0]	[31:28]	0100b