Intel® Open Source HD Graphics Programmers' Reference Manual (PRM)

Volume 2, Part 3: Command Reference - Registers

For the 2014 Intel Atom™ Processors, Celeron™ Processors, and Pentium™ Processors based on the "BayTrail" Platform (ValleyView graphics)

Creative Commons License

Notices and Disclaimers

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Implementations of the I2C bus/protocol may require licenses from various entities, including Philips Electronics N.V. and North American Philips Corporation.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Table of Contents

BBA_LEVEL2 - 2nd Level Batch Buffer Address

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	12144h

This register is to read the current value of the 2nd level batch buffer address. Since the 2nd level batch buffer logic is shared with the C6 work-around batch buffer, this also shows the work-around address when it is active.

DWord	Bit	Description
0	31:2	**WA Batch Buffer Address** Format: U30 Pointer to the WA Batch Buffer Address.
	1:0	**Reserved** Format: MBZ

ARB_MODE - Arbiter Mode Control Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04030h-04033h

ARB_MODE - Arbiter Mode Control register

DWord	Bit	Description
0	31:16	**Mask Bits** Default Value: 0000000000000000b Access: RO Mask bits Format: U16 Mask bits act as write enables for the bits in the lower bits of this register
	15	**Reserved**
	14	**GAM to Bypass GTT Translation** Default Value: 0b Access: R/W GAM to bypass GTT translation and pass logical addresses through with 0's padded on the MSBs to form the physical address
	13	**DC GDR** Default Value: 0b Access: R/W DC GDR
	12	**HIZ GDR** Default Value: 0b Access: R/W HIZ GDR
	11	**STC GDR** Default Value: 0b Access: R/W STC GDR

ARB_MODE - Arbiter Mode Control Register

	10	**BLB GDR** Default Value: 0b Access: R/W BLB GDR
	9	**GAM PD GDR** Default Value: 0b Access: R/W GAM PD GDR
	8	**Extra Register** Default Value: 0b Access: R/W Reserve Bit
	7:6	**Reserved**
	5	**Address Swizzling for Tiled Surfaces** Default Value: 0b Access: R/W Address Swizzling for Tiled-Surfaces Format: U1 This register location is updated via GFX Driver prior to enabling DRAM accesses. Driver needs to obtain the need for memory address swizzling via DRAM configuration registers and set the following bits (in Display Engine and Render/Media access streams) 0: No address Swizzling 1: Address bit[6] needs to be swizzled for tiled surfaces
	4	**VMC GDR Enable** Default Value: 0b

ARB_MODE - Arbiter Mode Control Register

		Access: R/W When this bit is set, Data requested from the VMC client will be generated by the GDR algorithm
	3	**Texture (MT) Cache GDR Enable bit** Default Value: 0b Access: R/W Texture Cache GDR enable bit Format: U1 When this bit is set, Data requested from the Texture Cache client will be generated by the GDR algorithm
	2	**Depth (RCZ) Cache GDR Enable bit** Default Value: 0b Access: R/W Depth Cache GDR enable bit Format: U1 When this bit is set, Data requested from the Depth Cache client will be generated by the GDR algorithm (See GDR algorithm in xxx section)
	1	**Color (RCC) Cache GDR Enable bit** Default Value: 0b Access: R/W Color Cache GDR enable bit Format: U1 When this bit is set, Data requested from the Color Cache client will be generated by the GDR algorithm (See GDR algorithm in xxx section)
	0	**GTT Accesses GDR** Default Value: 0b Access: R/W GTT Accesses GDR Format: U1

ARB_MODE - Arbiter Mode Control Register		
		When this bit is enabled along with the Client's GDR bit, PPGTT and GGTT requests for this memory access will also be tagged as GDR to SQ.

ARB_MODE - Arbiter Mode Control register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Size (in bits):	32
Trusted Type:	1
Address:	04030h

DWord	Bit	Description
0	31:16	**Mask Bits** Default Value: 0000000000000000b Access: RO Format: U16 Mask bits act as write enables for the bits in the lower bits of this register
	15	**Reserved**
	14	**GAM to Bypass GTT Translation (GAM2BGTTT)** Default Value: 0b Access: R/W Format: MBZ GAM to bypass GTT translation and pass logical addresses through with 0's padded on the MSBs to form the physical address.
	13	**DC GDR (DC_GDR)** Default Value: 0b Access: R/W DC GDR
	12	**HIZ GDR (HIZ_GDR)** Default Value: 0b Access: R/W HIZ GDR
	11	**STC GDR (STC_GDR)** Default Value: 0b Access: R/W Format: U1 STC GDR
	10	**BLB GDR (STC_GDR)**

ARB_MODE - Arbiter Mode Control register

<table>
<tr><td></td><td></td><td><table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>BLB GDR</td></tr>
<tr><td></td><td>9</td><td><b>GAM PD GDR (GAMPD_GDR)</b><table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>GAM PD GDR</td></tr>
<tr><td></td><td>8</td><td><b>Color/Depth Port Share Bit (CDPS)</b><table><tr><td>Default Value:</td><td>00b</td></tr><tr><td>Access:</td><td>R/W</td></tr><tr><td>Format:</td><td>U1</td></tr></table>Color/Depth port share bit
This bit is used to force Color and Depth Caches to share an arbiter read request port. By default (Bit = 0) the Color Cache will NOT share the read request port with the Depth Cache.</td></tr>
<tr><td></td><td>7:6</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>5</td><td><b>Address Swizzling for Tiled Surfaces (AS4TS)</b><table><tr><td>Access:</td><td>R/W</td></tr><tr><td>Format:</td><td>U1</td></tr></table>Address Swizzling for Tiled-Surfaces. This register location is updated via GFX Driver prior to enabling DRAM accesses. Driver needs to obtain the need for memory address swizzling via DRAM configuration registers and set the following bits (in Display Engine and Render/Media access streams).<table><tr><th>Value</th><th>Name</th></tr><tr><td>0b</td><td>No address Swizzling</td></tr><tr><td>1b</td><td>Address bit[6] needs to be swizzled for tiled surfaces</td></tr></table></td></tr>
<tr><td></td><td>4</td><td><b>VMC GDR Enable (VMC_GDR_EN)</b><table><tr><td>Access:</td><td>R/W</td></tr></table>When this bit is set, Data requested from the VMC client will be generated by the GDR algorithm</td></tr>
<tr><td></td><td>3</td><td><b>Texture Cache GDR Enable bit (TCGDREN)</b><table><tr><td>Access:</td><td>R/W</td></tr><tr><td>Format:</td><td>U1</td></tr></table>Texture Cache GDR enable bit When this bit is set, Data requested from the Texture Cache client will be generated by the GDR algorithm (See GDR algorithm in xxx section)</td></tr>
<tr><td></td><td>2</td><td><b>Depth Cache GDR enable bit (DCGDREN)</b></td></tr>
</table>

ARB_MODE - Arbiter Mode Control register

		Access: R/W Format: U1 When this bit is set, Data requested from the Depth Cache client will be generated by the GDR algorithm (See the GDR algorithm section)
	1	**Color Cache GDR enable bit(CCGDREN)** Access: R/W Format: U1 When this bit is set, Data requested from the Color Cache client will be generated by the GDR algorithm (See GDR algorithm in xxx section)
	0	**GTT Accesses GDR (GTTAGDR)** Default Value: 0b Access: R/W Format: U1 When this bit is enabled along with the Client's GDR bit, PPGTT and GGTT requests for this memory access will also be tagged as GDR to SQ.

3DPRIM_END_OFFSET - Auto Draw End Offset

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02420h-02423h

DWord	Bit	Description
0	31:0	**End Offset** Format: U32 This register is used to store the end offset value used by the Vertex Fetch to determine when to stop processing the 3D_PRIMITIVE command. This register is valid when the End Offset Enable is set in the 3D_PRIMITIVE command.

BB_ADDR - Batch Buffer Head Pointer Register

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	02140h
Name:	RCS Batch Buffer Head Pointer Register
ShortName:	RCS_BB_ADDR
Address:	12140h
Name:	VCS Batch Buffer Head Pointer Register
ShortName:	VCS_BB_ADDR
Address:	22140h
Name:	BCS Batch Buffer Head Pointer Register
ShortName:	BCS_BB_ADDR

This register contains the current DWord Graphics Memory Address of the last-initiated batch buffer.

Programming Notes
Programming Restriction: This register should NEVER be programmed by driver. This is for HW internal use only.

DWord	Bit	Description
0	31:3	**Batch Buffer Head Pointer** Source: BlitterCS, VideoCS Exists If: //BCS, VCS Format: GraphicsAddress[31:3] This field specifies the DWord-aligned Graphics Memory Address where the last initiated Batch Buffer is currently fetching commands. If no batch buffer is currently active, the Valid bit will be 0 and this field will be meaningless.
	31:2	**Batch Buffer Head Pointer** Source: RenderCS Exists If: //RCS Format: GraphicsAddress[31:2] This field specifies the DWord-aligned Graphics Memory Address where the last initiated Batch Buffer is currently fetching commands. If no batch buffer is currently active, the Valid bit will be 0 and this field will be meaningless.
	2	**Reserved** Source: BlitterCS, VideoCS

BB_ADDR - Batch Buffer Head Pointer Register

<table>
<tr><td></td><td></td><td><table><tr><td>Exists If:</td><td>//BCS, VCS</td></tr><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>1</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>0</td><td><b>Valid</b><table><tr><td>Format:</td><td>U1</td></tr></table><table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td>Invalid <b>[Default]</b></td><td>Batch buffer Invalid</td></tr><tr><td>1h</td><td>Valid</td><td>Batch buffer Valid</td></tr></table></td></tr>
</table>

BB_STATE - Batch Buffer State Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS, VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32

Address:	12110h
Name:	VCS Batch Buffer State Register
ShortName:	VCS_BB_STATE

Address:	22110h
Name:	BCS Batch Buffer State Register
ShortName:	BCS_BB_STATE

This register contains the attributes of the current batch buffer initiated from the Ring Buffer.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:7</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>6</td><td><b>2nd Level Buffer Security Indicator</b>
<table><tr><td>Source:</td><td>VideoCS</td></tr><tr><td>Exists If:</td><td>//VCS</td></tr></table>
If set, VCS is fetching 2nd level batch commands from a PPGTT address space. If clear, this batch buffer is secure and will be accessed via the GGTT. Note:
<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td>MIBUFFER_SECURE <b>[Default]</b></td><td>Located in GGTT memory</td></tr><tr><td>1h</td><td>MIBUFFER_NONSECURE</td><td>Located in PPGTT memory</td></tr></table></td></tr>
<tr><td></td><td>6</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>6</td><td><b>Reserved</b>
<table><tr><td>Source:</td><td>BlitterCS</td></tr><tr><td>Exists If:</td><td>//BCS</td></tr><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>5</td><td><b>1st Level Buffer Security Indicator</b>
<table><tr><td>Format:</td><td>MI_1stBufferSecurityType</td></tr></table>
If set, BCS is fetching 1st level batch commands from a PPGTT address space. If clear, this batch buffer is secure and will be accessed via the GGTT.
<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td>MIBUFFER_SECURE <b>[Default]</b></td><td>Located in GGTT memory</td></tr><tr><td>1h</td><td>MIBUFFER_NONSECURE</td><td>Located in PPGTT memory</td></tr></table></td></tr>
<tr><td></td><td>4</td><td><b>Reserved</b></td></tr>
</table>

BB_STATE - Batch Buffer State Register

		Source:	BlitterCS
		Exists If:	//BCS
		Format:	MBZ
	4	**Reserved**	
	3:0	**Reserved**	
		Format:	MBZ

BCS_CXT_SIZE - BCS Context Sizes

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000402
Access:	Read/32 bit Write Only
Size (in bits):	32
Address:	221A8h

DWord	Bit	Description
0	31:13	**Reserved** Format: MBZ
	12:8	**BCS Context Size** Format: U5 Value | Name 4h | **[Default]**
	7:5	**Reserved** Format: MBZ
	4:0	**Reserved**

BCS_EIR - BCS Error Identity Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	R/WC
Size (in bits):	32
Address:	220B0h

The EIR register contains the persistent values of Hardware-Detected Error Condition bits. Any bit set in this register will cause the Master Error bit in the ISR to be set. The EIR register is also used by software to clear detected errors (by writing a '1' to the appropriate bit(s) except for the unrecoverable bits described).).

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table>
</td></tr>
<tr><td></td><td>15:0</td><td><b>Error Identity Bits</b>
<table><tr><td>Format:</td><td>Array of Error condition bits See Table 1 5. Hardware-Detected Error Bits</td></tr></table>
This register contains the persistent values of ESR error status bits that are unmasked via the EMR register. The logical OR of all (defined) bits in this register is reported in the Master Error bit of the Interrupt Status Register. To clear an error condition, software must first clear the error by writing a '1' to the appropriate bit(s) in this field. If required, software should then proceed to clear the Master Error bit of the IIR.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td><b>[Default]</b></td><td></td></tr>
<tr><td>1h</td><td>Error occurred</td><td>Error occurred</td></tr>
</table>
<table>
<tr><th>Programming Notes</th></tr>
<tr><td>Writing a '1' to a set bit will cause that error condition to be cleared. However, the Instruction Error bit (Bit 0) cannot be cleared except by reset (i.e., it is a fatal error).</td></tr>
</table>
</td></tr>
</table>

BCS_EMR - BCS Error Mask Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x0000FFFF
Access:	R/W
Size (in bits):	32
Address:	220B4h

The EMR register is used by software to control which Error Status Register bits are "masked" or "unmasked". "Unmasked" bits will be reported in the EIR, thus setting the Master Error ISR bit and possibly triggering a CPU interrupt, and will persist in the EIR until cleared by software. "Masked" bits will not be reported in the EIR and therefore cannot generate Master Error conditions or CPU interrupts.

Undefined or reserved bits in the Hardware Detected Error Bit Table will always return a read value of '1'

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td rowspan="2">0</td><td>31:16</td><td><b>Reserved</b>
<table>
<tr><td>Default Value:</td><td>0000h</td></tr>
<tr><td>Format:</td><td>MBZ</td></tr>
</table></td></tr>
<tr><td>15:0</td><td><b>Error Mask Bits</b>
<table>
<tr><td>Format:</td><td>Array of error condition mask bits See Table 1 5. Hardware-Detected Error Bits</td></tr>
</table>
This register contains a bit mask that selects which error condition bits (from the ESR) are reported in the EIR.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0000h</td><td>Not Masked</td><td>Will be reported in the EIR</td></tr>
<tr><td>FFFFh</td><td>Masked <b>[Default]</b></td><td>Will not be reported in the EIR</td></tr>
</table></td></tr>
</table>

BCS_ESR - BCS Error Status Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	220B8h

The ESR register contains the current values of all Hardware-Detected Error condition bits (these are all by definition "persistent"). The EMR register selects which of these error conditions are reported in the persistent EIR (i.e., set bits must be cleared by software) and thereby causing a Master Error interrupt condition to be reported in the ISR.

DWord	Bit	Description
0	31:16	**Reserved** Format: MBZ
	15:0	**Error Status Bits** Format: Array of error condition bits See Table 1 5. Hardware-Detected Error Bits This register contains the non-persistent values of all hardware-detected error condition bits.

Value	Name	Description
0h	**[Default]**	
1h	Error Condition Detected	Error Condition detected

BCS_EXCC - BCS Execute Condition Code Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	R/W,RO
Size (in bits):	32
Trusted Type:	1
Address:	22028h

This register contains user defined and hardware generated conditions that are used by MI_WAIT_FOR_EVENT commands. An MI_WAIT_FOR_EVENT instruction excludes the executing ring from arbitration if the selected event evaluates to a 1, while instruction is discarded if the condition evaluates to a 0. Once excluded, a ring is enabled into arbitration when the selected condition evaluates to a 0. This register also contains control for the invalidation of indirect state pointers on context restore.

DWord	Bit	Description
0	31:16	**Mask Bits** Format: Mask[15:0] These bits serves as a write enable for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.
	15	**Reserved** Format: MBZ
	14:12	**Reserved** Format: MBZ
	11:5	**Reserved** Format: MBZ
	4:0	**User Defined Condition Codes** Format: U5 The software may signal a Stream Semaphore by setting the Mask bit and Signal Bit together to match the bit field specified in a WAIT_FOR_EVENT (Semaphore).

BCS_HWSTAM - BCS Hardware Status Mask Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0xFFFFFFFF
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	22098h

Access: RO for Reserved Control bits

The HWSTAM register has the same format as the Interrupt Control Registers. The bits in this register are "mask" bits that prevent the corresponding bits in the Interrupt Status Register from generating a "Hardware Status Write" (PCI write cycle). Any unmasked interrupt bit (HWSTAM bit set to 0) will allow the Interrupt Status Register to be written to the ISR location (within the memory page specified by the Hardware Status Page Address Register) when that Interrupt Status Register bit changes state.

Programming Notes

To write the interrupt to the HWSP, the corresponding IMR bit must also be clear (enabled). At most 1 bit can be unmasked at any given time.

DWord	Bit	Description
0	31:0	**Hardware Status Mask Register** Default Value: FFFFFFFFh Format: Array of Masks refer to Table 5-1 in Interrupt Control Register section for bit definitions

BCS_PWRCTX_MAXCNT - BCS IDLE Max Count

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000040
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	22054h

This register contains the time in 0.64us to wait before telling power management hardware the render pipe is IDLE

DWord	Bit	Description
0	31:20	**Reserved** Format: MBZ
	19:0	**Blitter IDLE Wait Time** Format: Max Count Specifies how long the command stream should wait before ensuring the pipe is IDLE and to let power management hardware know

Value	Name	Description
00040h	**[Default]**	0x00040 * 0.64us ~ 41us wait time

Programming Notes

- This is only useable if bit 0 of the PC_PSMI_CTRL is clear.
- The value in this field must be greater than 1.

BCS_INSTPM - BCS Instruction Parser Mode Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	220C0h

Desc

DWord	Bit	Description
0	31:16	**Mask Bits** Format: Mask[15:0] Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)
	15:11	**Reserved** Format: MBZ
	10	**Reserved** Format: MBZ
	9	**TLB Invalidate** Format: U1 If set, this bit allows the command stream engine to invalidate the blitter TLBs. This bit is valid only with the Sync flush enable **Note: GFX soft resets do not invalidate TLBs, it is up to GFX driver to explicitly invalidate TLBs post reset.**
	8:7	**Reserved** Format: MBZ
	6	**Memory Sync Enable** Format: U1 This set, this bit allows the blitter decode engine to write out the data from the local caches to memory. This bit is not persistent. S/W must define this bit each time a sync flush is requested
	5	**Sync Flush Enable** Format: U1 Format: Enable Cleared by HW This field is used to request a Sync Flush operation. The device will automatically clear this bit before completing the operation. See Sync Flush (Programming Environment).
	4:0	**Reserved** Format: MBZ

BCS_IMR - BCS Interrupt Mask Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0xFFFFFFFF
Access:	R/W
Size (in bits):	32
Address:	220A8h

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

DWord	Bit	Description
0	31:0	**Interrupt Mask Bits** Format: Array of interrupt mask bits Refer to Table 5-1 in Interrupt Control Register section for bit definitions This field contains a bit mask which selects which interrupt bits (from the ISR) are reported in the IIR.

Value	Name	Description
FFFF FFFFh	**[Default]**	
0h	Not Masked	Will be reported in the IIR
1h	Masked	Will not be reported in the IIR

BCS_MI_MODE - BCS Mode Register for Software Interface

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000200
Access:	R/W
Size (in bits):	32
Address:	2209Ch-2209Fh

The MI_MODE register contains information that controls software interface aspects of the command parser.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Masks</b>
A 1 in a bit in this field allows the modification of the corresponding bit in Bits 15:0</td></tr>
<tr><td></td><td>15</td><td><b>Suspend Flush</b>
Mask: MMIO(0x209c)#31
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td>No Delay</td><td>HW will not delay flush, this bit will get cleared by MI_SUSPEND_FLUSH as well</td></tr>
<tr><td>1h</td><td>Delay Flush</td><td>Suspend flush is active</td></tr>
</table></td></tr>
<tr><td></td><td>14:12</td><td><b>Reserved</b>
Read/Write</td></tr>
<tr><td></td><td>11</td><td><b>Invalidate UHPTR enable</b>
If bit set H/W clears the valid bit of BCS_UHPTR (4134h, bit 0) when current active head pointer is equal to UHPTR.</td></tr>
<tr><td></td><td>10</td><td><b>Reserved</b>
<table>
<tr><td>Format:</td><td>MBZ</td></tr>
</table></td></tr>
<tr><td></td><td>9</td><td><b>Ring Idle (Read Only Status Bit)</b>
<i>Writes to this bit are not allowed.</i>
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0</td><td>Parser not idle</td></tr>
<tr><td>1</td><td>Parser idle <b>[Default]</b></td></tr>
</table></td></tr>
<tr><td></td><td>8</td><td><b>Stop Ring</b>
Software must set this bit to force the Ring and Command Parser to Idle. Software must read a 1 in Ring Idle bit after setting this bit to ensure that the hardware is idle.
<i>Software must clear this bit for Ring to resume normal operation.</i>
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0</td><td>Normal Operation</td></tr>
<tr><td>1</td><td>Parser is turned off</td></tr>
</table></td></tr>
<tr><td></td><td>7:2</td><td><b>Reserved</b></td></tr>
</table>

BCS_MI_MODE - BCS Mode Register for Software Interface

<table>
<tr><td></td><td></td><td>Read/Write</td></tr>
<tr><td></td><td>1</td><td><b>Bypass Fence Write</b>
If set, this bit will bypass all writes during flushes, independent of programming. This includes post-sync op bits, the implicit TLB invalidate write (set in GFX_MODE[13]), and sync flush fences.
<i>Note this is only intended for work-arounds</i>
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0</td><td>Normal Operation</td></tr>
<tr><td>1</td><td>Bypass</td></tr>
</table>
</td></tr>
<tr><td></td><td>0</td><td><b>Reserved</b>
Read/Write</td></tr>
</table>

BCS_PP_DCLV - BCS PPGTT Directory Cacheline Valid Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Address:	22220h

Default Value = 0h

This register controls update of the on-chip PPGTT Directory Cache during a context restore. Bits that are set will trigger the load of the corresponding 16 directory entry group. This register is restored with context (prior to restoring the on-chip directory cache itself). This register is also restored when switching to a context whose LRCA matches the current CCID if the Force PD Restore bit is set in the context descriptor.

The context image of this register must be updated and maintained by SW; SW should not normally need to read this register.

This register can also effectively be used to limit the size of a processes' virtual address space. Any access by a process that requires a PD entry in a set that is not enabled in this register will cause a fatal error, and no fetch of the PD entry will be attempted.

DWord	Bit	Description
0	63:32	**Reserved** Format: MBZ
	31:0	**PPGTT Directory Cache Restore** Format: Enable[32] [1..32] 16 entries If set, the [1st..32nd] 16 entries of the directory cache are considered valid and will be brought in on context restore. If clear, these entries are considered invalid and fetch of these entries will not be attempted.

BCS_TLBPEND_RDY0 - BCS Ready Bit Vector for TLBPEND Registers

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24708h-2470Bh

This register contains the ready bits for entries 0-31 of TLBPEND structure (Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Ready bits per entry**

BCS_TIMESTAMP - BCS Reported Timestamp Count

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000, 0x00000000
Access:	RO. This register is not set by the context restore.
Size (in bits):	64
Address:	22358h

This register provides an elapsed real-time value that can be used as a timestamp. This register is not reset by a graphics reset. It will maintain its value unless a full chipset reset is performed.

Note: This timestamp register reflects the value of the PCU TSC. The PCU TSC counts 10ns increments; this timestamp reflects bits 38:3 of the TSC (i.e. 80ns granularity, rolling over every 1.5 hours).

DWord	Bit	Description
0	63:36	**Reserved** Format: MBZ
	35:0	**Timestamp Value** Format: U36 This register toggles every 80 ns. The upper 28 bits are zero.

BCS_RNCID - BCS Ring Buffer Next Context ID Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Address:	22198h-2219Fh

This register contains the *next* ring context ID associated with the ring buffer.

Programming Notes
The current context (RCCID) register can be updated indirectly from this register on a context switch event. Note that the only time a context switch can occur is when MI_ARB_CHECK enables preemption or the current context runs dry (head pointer becomes equal to tail pointer).

DWord	Bit	Description
0	63:0	**Unnamed** See Context Descriptor for BCS

BCS_TLBPEND_SEC0 - BCS Section 0 of TLBPEND entry

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24400h-24403h

This register is directly mapped to the TLBPEND Array in the Graphic Arbiter.

DWord	Bit	Description
0	31:28	**GTT bits** Bits 3:0 of the GTT entry used to translate the Virtual Address. 000 if translation is pending.
	27:0	**Current Address** The value of this field depends on the stage of the TLB translation for this entry: VA - bits 27:20 = 00, bits 19:0 = Bits 31:12 of the Virtual Address of the cycle.

BCS_TLBPEND_SEC1 - BCS Section 1 of TLBPEND entry

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24500h-24503h

This register is directly mapped to the current Virtual Addresses in the MTTLB (Texture and constant cache TLB).

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31</td><td><b>vtstatus</b>
This bit will be used in conjunction with the ready bit to determine the stage of the translation. See table in section 0 register.</td></tr>
<tr><td></td><td>30:28</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>27:24</td><td><b>PAT entry</b>
Location of Physical Address in Physical Address Table.</td></tr>
<tr><td></td><td>23:22</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>21:20</td><td><b>Surface format</b>
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0xb</td><td>Linear</td></tr>
<tr><td>10b</td><td>Tile X</td></tr>
<tr><td>11b</td><td>Tile Y</td></tr>
</table></td></tr>
<tr><td></td><td>19:14</td><td><b>Cache line offset in page</b></td></tr>
<tr><td></td><td>13:10</td><td><b>Cacheability Control Bits</b></td></tr>
<tr><td></td><td>9</td><td><b>ZLR bit</b>
indicates a zero length read</td></tr>
<tr><td></td><td>8:2</td><td><b>TAG</b></td></tr>
<tr><td></td><td>1:0</td><td><b>SRC ID</b>
00/01=BCS; 10/11= BLB</td></tr>
</table>

BCS_PSMI_CTRL - BCS Sleep State and PSMI Control

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	22050h

This register is to be used to control all aspects of PSMI and power saving functions

DWord	Bit	Description
0	31:16	**Mask Bits** Format: Mask[15:0] Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)
	15	**Reserved** Format: MBZ
	14:13	**Reserved** Format: MBZ
	12	**Reserved** Format: MBZ
	11:8	**Reserved** Format: MBZ
	7	**Reserved** Format: MBZ
	6:5	**Reserved** Format: MBZ
	4	**GO Indicator** Access: RO Format: GO This is a read only field. Writing to this bit is undefined. To simplify power saving and soft reset flows, the power management hardware has the ability to block all pending memory cycles of the render pipe. When GO=0, all cycles are blocked. All CPD enter/exit and RC6 enter/exit has this bit set to 0.

Value	Name	Description
0h	Disable **[Default]**	No cycles allowed coming out of IDLE. All pending memory read cycles are complete. No new cycles permitted except for power context or PSMI cycles.
1h	Enable	Normal execution

<table>
<tr><th colspan="3">BCS_PSMI_CTRL - BCS Sleep State and PSMI Control</th></tr>
<tr><td></td><td>3</td><td><b>IDLE Indicator</b><table><tr><td>Default Value:</td><td>0h Render is assumed NOT IDLE coming out of reset</td></tr><tr><td>Access:</td><td>RO</td></tr><tr><td>Format:</td><td>IDLE</td></tr></table>This is a read only field. Writing to this bit is undefined. This indicates what power management thinks what state the render pipe is in. That is, if set, the full handshake between render and power management has occured and most likely the render clocks are currently turned off.</td></tr>
<tr><td></td><td>2</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>1</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>0</td><td><b>Reserved</b></td></tr>
</table>

BCS_SWCTRL - BCS SW Control

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	22200h

DWord	Bit	Description
0	31:16	**Masks** Format: U16 A "1" in a bit in this field allows the modification of the corresponding bit in bits 15:0.
	15:4	**Reserved** Format: MBZ
	3:2	**Reserved** Format: MBZ
	1	**Tile Y Destination** Format: U1 Programming this bit makes the HW treat all destination surfaces as Tile Y. This bit over-rides the setting of the destination format in the packet provided to the blitter command streamer. SW is required to flush the HW before changing the polarity of this bit. This bit is part of the context save/restore.
	0	**Tile Y Source** Format: U1 Programming this bit makes the HW treat all source surfaces as Tile Y. This bit over-rides the setting of the source format in the packet provided to the blitter command streamer. SW is required to flush the HW before changing the polarity of this bit. This bit is part of the context save/restore.

BCSTLB_VA - BCS TLB Virtual Page Address Registers

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24800h-24803h

This register is directly mapped to the current Virtual Addresses in the BCS TLB.

DWord	Bit	Description
0	31:12	**ADDRESS** Format: GraphicsAddress[31:12] PAGE VIRTUAL ADDRESS.
	11:0	**RESERVED** Format: MBZ

BCS_TLBPEND_VLD0 - BCS Valid Bit Vector for TLBPEND registers

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24700h-24703h

This register contains the valid bits for entries 0-31 of TLBPEND structure(Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

BCS_SYNC_FLIP_STATUS - BCS Wait for event and Display flip flags Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	222D0h

This register is the saved value of what wait for events are still valid. This register is part of context save and restore for RC6 feature.

DWord	Bit	Description
0	31	**Reserved** Format: MBZ
	30	**Display Plane A Asynchronous Display Flip Pending** Format: Enable This field enables a wait for the duration of a Display Plane A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	29	**Display Plane A Synchronous Flip Display Pending** Format: Enable This field enables a wait for the duration of a Display Plane A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	28	**Display Sprite A Synchronous Flip Display Pending** Format: Enable This field enables a wait for the duration of a Display Sprite A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
	27	**Reserved** Format: MBZ
	26	**Display Plane B Asynchronous Display Flip Pending** Format: Enable This field enables a wait for the duration of a Display Plane B "Flip Pending" condition. If a flip

BCS_SYNC_FLIP_STATUS - BCS Wait for event and Display flip flags Register

		request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	25	**Display Plane B Synchronous Flip Display Pending** Format: | Enable This field enables a wait for the duration of a Display Plane B "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	24	**Display Sprite B Synchronous Flip Display Pending** Format: | Enable This field enables a wait for the duration of a Display Sprite B "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
	23	**Reserved** Format: | MBZ
	22	**Display Plane A Asynchronous Flip Pending Wait Enable** Format: | Enable This field enables a wait for the duration of a Display Plane A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	21	**Display Plane A Synchronous Flip Pending Wait Enable** Format: | Enable This field enables a wait for the duration of a Display Plane A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	20	**Display Sprite A Synchronous Flip Pending Wait Enable** Format: | Enable This field enables a wait for the duration of a Display Sprite A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.

BCS_SYNC_FLIP_STATUS - BCS Wait for event and Display flip flags Register

<table>
<tr><td></td><td>19:15</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>14</td><td><b>Display Plane B Asynchronous Flip Pending Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait for the duration of a Display Plane B "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>13</td><td><b>Display Plane B Synchronous Flip Pending Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait for the duration of a Display Plane B "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>12</td><td><b>Display Sprite B Synchronous Flip Pending Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait for the duration of a Display Sprite B "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>11</td><td><b>Display Plane C Asynchronous Display Flip Pending</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait for the duration of a Display Plane C "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>10</td><td><b>Display Plane C Synchronous Display Flip Pending</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait for the duration of a Display Plane C "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>9</td><td><b>Display Sprite C Synchronous Flip Display Pending</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait for the duration of a Display Sprite C "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front</td></tr>
</table>

BCS_SYNC_FLIP_STATUS - BCS Wait for event and Display flip flags Register

<table>
<tr><td></td><td></td><td>buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>8</td><td><b>Display Plane C Asynchronous Flip Pending Wait Enable</b>
<table><tr><td>Format:</td><td>Enable</td></tr></table>
This field enables a wait for the duration of a Display Plane C "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>7</td><td><b>Display Plane C Synchronous Flip Pending Wait Enable</b>
<table><tr><td>Format:</td><td>Enable</td></tr></table>
This field enables a wait for the duration of a Display Plane C "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>6</td><td><b>Display Sprite C Synchronous Flip Pending Wait Enable</b>
<table><tr><td>Format:</td><td>Enable</td></tr></table>
This field enables a wait for the duration of a Display Sprite C "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions..</td></tr>
<tr><td></td><td>5</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>4:0</td><td><b>Condition Code Wait Select</b>
This field enables a wait for the duration that the corresponding condition code is active. These enable select one of 15 condition codes in the EXCC register, that cause the parser to wait until that condition-code in the EXCC is cleared.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td>Not Enabled</td><td>Condition Code Wait not enabled</td></tr>
<tr><td>1h-5h</td><td>Enabled</td><td>Condition Code select enabled; selects one of 5 codes, 0 - 4</td></tr>
<tr><td>6h-15h</td><td>Reserved</td><td></td></tr>
</table>
<table>
<tr><th>Programming Notes</th></tr>
<tr><td>Note that not all condition codes are implemented. The parser operation is UNDEFINED if an unimplemented condition code is selected by this field. The description of the EXCC register (Memory Interface Registers) lists the codes that are implemented.</td></tr>
</table></td></tr>
</table>

BCS_CTR_THRSH - BCS Watchdog Counter Threshold

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00150000
Access:	R/W
Size (in bits):	32
Address:	2217Ch

DWord	Bit	Description
0	31:0	**Counter logic Threshold** Default Value: 00150000h Format: U32 This field specifies the threshold that the hardware checks against for the value of the blitter clock counter before generating an interrupt. The counter in hardware generates an interrupt when the threshold is reached, rolls over and starts counting again. The interrupt generated is the "Media Hang Notify" interrupt since this watchdog timer is intended primarily to remedy VLD hangs on the main pipeline.

MFC_BITSTREAM_SE_BITCOUNT_SLICE - Bitstream Output Bit Count for the last Syntax Element Report Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124D4h
Name:	VDBOX1

This register stores the count of number of bits in the bitstream for the last syntax element before padding. The bit count is before the byte-aligned alignment padding insertion, but includes the stop-one-bit. This register is part of the context save and restore.

DWord	Bit	Description
0	31:0	**MFC Bitstream Syntax Element Bit Count** Total number of bits in the bitstream output before padding. This count is updated each time the internal counter is incremented.

MFC_BITSTREAM_BYTECOUNT_SLICE - Bitstream Output Byte Count Per Slice Report Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124D0h

This register stores the count of bytes of the bitstream output. This register is part of the context save and restore.

DWord	Bit	Description
0	31:0	**MFC Bitstream Byte Count** Total number of bytes in the bitstream output from the encoder. This count is updated for every time the internal bitstream counter is incremented.

MFC_AVC_MINSIZE_PADDING_COUNT - Bitstream Output Minimal Size Padding Count Report Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12414h
Name:	VDBOX1

This register stores the count in bytes of **minimal size padding insertion. It is primarily provided for statistical data gathering.** This register is part of the context save and restore.

DWord	Bit	Description
0	31:0	**MFC AVC MinSize Padding Count** Total number of bytes in the bitstream output contributing to minimal size padding operation. This count is updated each time when the padding count is incremented.

BLBTLB_VA - BLBTLB_VA Virtual page Address Registers

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24900h-24903h

This register is directly mapped to the current Virtual Addresses in the BLB TLB.

DWord	Bit	Description
0	31:12	**ADDRESS** Format: GraphicsAddress[31:12] PAGE VIRTUAL ADDRESS
	11:0	**RESERVED** Format: MBZ

BLT_ENG_FR - Blitter Engine Fault Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04294h-04297h

Blitter Engine Fault Register

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:12</td><td><b>Virtual Address of Page Fault</b>
<table><tr><td>Default Value:</td><td>00000h</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>
This is the original Address of the Page that generated the First fault for this engine.

This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW</td></tr>
<tr><td></td><td>11</td><td><b>Blitter GTTSEL</b>
<table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>
This bit indicates if the valid bit happened while using PPGTT or GGTT: 0 - PPGTT, 1 - GGTT

This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW</td></tr>
<tr><td></td><td>10:3</td><td><b>SRCID of Fault</b>
<table><tr><td>Default Value:</td><td>00h</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>
This is the Source ID of the unit that requested the cycle that generated the First Page fault for this engine.

This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW</td></tr>
<tr><td></td><td>2:1</td><td><b>Fault Type</b>
<table><tr><td>Default Value:</td><td>00b</td></tr></table></td></tr>
</table>

BLT_ENG_FR - Blitter Engine Fault Register

<table>
<tr>
<td></td>
<td></td>
<td>
<table><tr><td>Access:</td><td>R/W</td></tr></table>
Type of Fault recorded:

00 - Page Fault.

01 - Invalid PD Fault

10 - Unloaded PD Fault

11 - Invalid and Unloaded PD fault

This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW
</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>
<b>Valid Bit</b>
<table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>
This bit indicates that the first fault for this engine has been recorded. It can only be cleared by SW, which will also clear the other fields.
</td>
</tr>
</table>

BLT_MODE - Blitter Mode Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	2229Ch

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Mask Bits</b><table><tr><td>Format:</td><td>Mask[15:0]</td></tr></table>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td></tr>
<tr><td></td><td>15:14</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>13:10</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>9</td><td><b>Per-Process GTT Enable</b><table><tr><td>Format:</td><td>Enable Per-Process GTT BS Mode Enable</td></tr></table><table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td>PPGTT Disable <b>[Default]</b></td><td>When clear, the Global GTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td></tr><tr><td>1h</td><td>PPGTT Enable</td><td>When set, the PPGTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td></tr></table></td></tr>
<tr><td></td><td>8</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>7:4</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>3:1</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>0</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
</table>

BRSYNC - Blitter/Render Semaphore Sync Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	22040h

This register is written by CS, read by BCS.

DWord	Bit	Description
0	31:0	**Semaphore Data** Semaphore data for synchronization between blitter engine and render engine.

BVSYNC - Blitter/Video Semaphore Sync Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	22044h

This register is written by VCS, read by BCS.

DWord	Bit	Description
0	31:0	**Semaphore Data** Semaphore data for synchronization between blitter engine and video codec engine.

CACHE_MODE_0 - Cache Mode Register 0

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000004
Access:	R/W
Size (in bits):	32
Address:	07000h

Description
This register is used to control the operation of the Render and Sampler L2 Caches. All reserved bits are implemented as read/write. Before changing the value of this register, GFX pipeline must be idle i.e. full flush is required. This Register is saved and restored as part of Context.
RegisterType = MMIO_SVL

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Masks</b>
<table><tr><td>Format:</td><td>Mask[15:0]</td></tr></table>
A 1 in a bit in this field allows the modification of the corresponding bit in Bits 15:0.</td></tr>
<tr><td></td><td>15</td><td><b>Sampler L2 Disable</b>
<table><tr><td>Format:</td><td>Disable</td></tr></table>
<table><tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td><b>[Default]</b></td><td>Sampler L2 Cache Enabled.</td></tr>
<tr><td>1h</td><td></td><td>Sampler L2 Cache Disabled. All accesses are treated as misses.</td></tr></table></td></tr>
<tr><td></td><td>14:12</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>11</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>10</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>9</td><td><b>Sampler L2 TLB Prefetch Enable</b>
<table><tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td><b>[Default]</b></td><td>TLB Prefetch Disabled</td></tr>
<tr><td>1h</td><td></td><td>TLB Prefetch Enabled</td></tr></table></td></tr>
<tr><td></td><td>8</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>7:6</td><td><b>Sampler L2 Request Arbitration</b>
<table><tr><td>Format:</td><td>U2</td></tr></table></td></tr>
</table>

CACHE_MODE_0 - Cache Mode Register 0

Value	Name	Description
00b		Round Robin
01b		Fetch are Highest Priority
10b		Constants are Highest Priority
11b		Reserved

5

STC Eviction Policy

Format:	Disable

If this bit is set, STCunit will have LRA as replacement policy. The default value i.e. (when this bit is reset) indicates that non-LRA eviction policy. This bit must be reset. LRA replacement policy is not supported.

Programming Notes
If this bit is set to "1", bit 4 of 0x7010h must also be set to "1".

4

RCC Eviction Policy

Format:	Disable

If this bit is set, RCCunit will have LRA as replacement policy. The default value i.e. (when this bit is reset) indicates that non-LRA eviction policy. This bit must be reset. LRA replacement policy is not supported.

Programming Notes
If this bit is set to "1", bit 7 of 0x7010h must also be set to "1".

3

Reserved

2

Hierarchical Z RAW Stall Optimization Disable

Format:	U1

The Hierarchical Z RAW Stall Optimization allows non-overlapping polygons in the same 8x4 pixel/sample area to be processed without stalling waiting for the earlier ones to write to Hierarchical Z buffer.

Value	Name	Description
0h	Enable	Enables the hierarchical Z RAW Stall Optimization.
1h	Disable **[Default]**	Disables the hierarchical Z RAW Stall Optimization.

1

Disable clock gating in the pixel backend

Format:	Disable

MCL related clock gating is disabled in the pixel backend. Before setting this bit to 1, the instruction/state caches must be invalidated.

0

Render Cache Operational Flush Enable

Format:	Enable

Value	Name	Description
0h	Disable	Operational Flush Disabled (recommended for performance when not

CACHE_MODE_0 - Cache Mode Register 0

			[Default]	rendering to the front buffer). Restriction: This bit must be set to '0' (Disable).
		1h	Enable	Operational Flush Enabled (required when rendering to the front buffer). Restriction: Do not use this value; this bit must be 0.

CACHE_MODE_1 - Cache Mode Register 1

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000180
Access:	Read/32 bit Write Only
Size (in bits):	32
Address:	07004h

Description
RegisterType: MMIO_SVL
Before changing the value of this register, GFX pipeline must be idle; i.e., full flush is required. This Register is saved and restored as part of Context.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Mask Bits for 15:0</b><table><tr><td>Format:</td><td>Mask[15:0]</td></tr></table>Must be set to modify corresponding data bit. Reads to this field returns zero.</td></tr>
<tr><td></td><td>15</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>14</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>13</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>12</td><td><b>HIZ Eviction Policy</b><table><tr><td>Format:</td><td>U1</td></tr></table>If this bit is set, Hizunit will have LRA as replacement policy. The default value i.e. (when this bit is reset) indicates the non-LRA eviction policy. For performance reasons, this bit must be reset.<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td><b>[Default]</b></td><td>Non-LRA eviction Policy</td></tr><tr><td>1h</td><td></td><td>LRA eviction Policy</td></tr></table><table><tr><th>Programming Notes</th></tr><tr><td>If this bit is set to "1", bit 3 of 0x7010h must also be set to "1"</td></tr></table></td></tr>
<tr><td></td><td>11</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>10</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>9</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>8:7</td><td><b>Sampler Cache Set XOR selection</b><table><tr><td>Format:</td><td>U2</td></tr></table>These bits have an impact only when the Sampler cache is configured in 16 way set associative</td></tr>
</table>

CACHE_MODE_1 - Cache Mode Register 1

mode. If the cache is being used for immediate data or for blitter data these bits have no effect.

Value	Name	Description
00b	None	No XOR.
01b	Scheme 1	New_set_mask[3:0] = Tiled_address[16:13]. New_set[3:0] less than or = New_set_mask[3:0] ^Old_set[3:0]. Rationale: These bits can distinguish among 16 different equivalent classes of virtual pages. These bits also represent the lsb for tile rows ranging from a pitch of 1 tile to 16 tiles.

10b	Scheme 2	New_set_mask[3] = Tiled_address[17] ^ Tiled_address[16].
		New_set_mask[2] = Tiled_address[16] ^ Tiled_address[15].
		New_set_mask[1] = Tiled_address[15] ^ Tiled_address[14].
		New_set_mask[0] = Tiled_address[14] ^ Tiled_address[13].
		New_set[3:0] less than or = New_set_mask[3:0] ^ Old_set[3:0].
		Rationale: More bits on each XOR can give better statistical uniformity on sets and since two lsbs are taken for each tile row size, it reduces the chance of aliasing on sets.

11b	Scheme 3 **[Default]**	New_set_mask[3] = Tiled_address[22] ^ Tiled_address[21] ^ Tiled_address[20] ^ Tiled_address[19].
		New_set_mask[2] = Tiled_address[18] ^ Tiled_address[17] ^ Tiled_address[16].
		New_set_mask[1] = Tiled_address[15] ^ Tiled_address[14].

CACHE_MODE_1 - Cache Mode Register 1

		New_set_mask[0] = Tiled_address[13]. New_set[3:0] less than or = New_set_mask[3:0] ^ Old_set[3:0]. Rationale: More bits on each XOR can give better statistical uniformity on sets and since each XOR has different bits, it reduces the chance of aliasing on sets even more.

6 **Pixel Backend sub-span collection Optimization Disable**

Format:	Disable

Value	Name	Description
0h	[Default]	Enables two contiguous quads to be collected as 4X2 access for RCZ interface. This allows for less bank collision and less RAM power on RCZ.
1h		Disables this optimization and therefore only one valid sub-span is sent to RCZ on the 4X2 interface.

Programming Notes
This bit must be set.

5 **MCS Cache Disable**

Format:	Disable

For Programming restrictions please refer to the 3D Pipeline.

Value	Name	Description
0h	[Default]	MCS cache enabled. It allows RTs with MCS buffer enabled to be rendered using either MSAA compression for MSRT OR with color clear feature for non MSRT.
1h		MCS cache is disabled. Hence no MSAA compression for MSRT and no color clear for non-MSRT.

4 **Reserved**

Format:	MBZ

3 **Depth Read Hit Write-Only Optimization Disable**

Format:	Disable

Description
This bit must always be reset to "0".

Value	Name	Description
0h		Read Hit Write-only optimization is enabled in the Depth cache

CACHE_MODE_1 - Cache Mode Register 1

<table>
<tr><td rowspan="4"></td><td></td><td><table><tr><td></td><td><b>[Default]</b></td><td>(RCZ).</td></tr><tr><td>1h</td><td></td><td>Read Hit Write-only optimization is disabled in the Depth cache (RCZ).</td></tr></table></td></tr>
<tr><td>2</td><td><b>Reserved</b></td></tr>
<tr><td>1</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td>0</td><td><b>Reserved</b></td></tr>
</table>

CASC_LRA_0 - CASC LRA 0

Register Space:	MMIO: 0/2/0
Default Value:	0x1F100F00
Size (in bits):	32
Address:	04060h-04063h

CASC LRA 0

DWord	Bit	Description
0	31:24	**CASC LRA1 Max** Default Value: 00011111b Access: R/W CASC LRA1 Max Format: U6 Maximum value of programmable LRA1 Maximum Allow Value: 159
	23:16	**CASC LRA1 Min** Default Value: 00010000b Access: R/W CASC LRA1 Min Format: U6 Minimum value of programmable LRA1
	15:8	**CASC LRA0 Max** Default Value: 00001111b Access: R/W CASC LRA0 Max Format: U6 Maximum value of programmable LRA0 Maximum Allow Value: 159
	7:0	**CASC LRA0 Min** Default Value: 00000000b Access: R/W CASC LRA0 Min

CASC_LRA_0 - CASC LRA 0		
		Format: U6 Minimum value of programmable LRA1

CASC_LRA_1 - CASC LRA 1

Register Space:	MMIO: 0/2/0
Default Value:	0x3F302F20
Size (in bits):	32
Address:	04064h-04067h

CASC LRA 1

DWord	Bit	Description
0	31:24	**CASC LRA3 Max** Default Value: 00111111b Access: R/W CASC LRA3 Max Format: U6 Maximum value of programmable LRA3
	23:16	**CASC LRA3 Min** Default Value: 00110000b Access: R/W CASC LRA3 Min Format: U6 Minimum value of programmable LRA3
	15:8	**CASC LRA2 Max** Default Value: 00101111b Access: R/W CASC LRA2 Max Format: U6 Maximum value of programmable LRA2
	7:0	**CASC LRA2 Min** Default Value: 00100000b Access: R/W CASC LRA2 Min Format: U6 Minimum value of programmable LRA2

CASC_LRA_2 - CASC LRA 2

Register Space:	MMIO: 0/2/0
Default Value:	0x00009F40
Size (in bits):	32
Address:	04068h-0406Bh

CASC LRA 2

DWord	Bit	Description
0	31:16	**Reserved** Default Value: 0000h Access: RO Reserved Format: MBZ
	15:8	**CASC LRA4 Max** Default Value: 10011111b Access: R/W CASC LRA4 Max Format: U6 Maximum value of programmable LRA4 Maximum Allow Value: 159
	7:0	**CASC LRA4 Min** Default Value: 01000000b Access: R/W CASC LRA4 Min Format: U6 Minimum value of programmable LRA4

CASC_LRA_3 - CASC LRA 3

Register Space:	MMIO: 0/2/0
Default Value:	0x000014E4
Size (in bits):	32
Address:	0406Ch-0406Fh

CASC LRA 3

DWord	Bit	Description
0	31:18	**Reserved** Default Value: 00000000000000b Access: RO Reserved Format: MBZ
	17:15	**BCS LRA** Default Value: 000b Access: R/W BCS LRA Format: U6 Which LRA should use
	14:12	**BLB LRA** Default Value: 001b Access: R/W BLB LRA Format: U6 Which LRA should use
	11:9	**VCS LRA** Default Value: 010b Access: R/W VCS LRA Format: U6 Which LRA should use
	8:6	**VMX LRA** Default Value: 011b

CASC_LRA_3 - CASC LRA 3

		Access: R/W VMX LRA Format: U6 Which LRA should use
	5:3	**VMC LRA** Default Value: 100b Access: R/W VMC LRA Format: U6 Which LRA should use
	2:0	**VCR LRA** Default Value: 100b Access: R/W VCR LRA Format: U6 Which LRA should use

CL_INVOCATION_COUNT - Clipper Invocation Counter

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02338h

This register stores the count of objects entering the Clipper stage. This register is part of the context save and restore.

DWord	Bit	Description
0	63:0	**CL Invocation Count Report** Number of objects entering the clipper stage. Updated only when Statistics Enable is set in CLIP_STATE (see the Clipper Chapter in the 3D Volume.)

CL_PRIMITIVES_COUNT - Clipper Primitives Counter

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02340h

This register reflects the total number of primitives that have been output by the clipper. This register is part of the context save and restore.

DWord	Bit	Description
0	63:0	**Clipped Primitives Output Count** Total number of primitives output by the clipper stage. This count is updated for every primitive output by the clipper stage, as long as Statistics Enable is set in SF_STATE (see the Clipper and SF Chapters in the 3D Volume.)

CGMSG - Clock Gating Messages

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	08104h

Clock Gating Messages Register

DWord	Bit	Description
0	31:16	**Message Mask** Access: RO Message Mask In order to write to bits 15:0, the corresponding message mask bits must be written. For example, for bit 14 to be set, bit 30 needs to be 1: 40004000
	15:2	**Reserved** Access: RO Reserved
	1	**Media 1 Clock gating control message** Access: R/W Gate Media 1 (2nd Vbox) Clock Message : '0': Media 1 Clock Un-gate Request (un-gates the cmclk clock in the 2n Media block) '1': Media 1 Clock Gate Request (gates the cmclk clockin the 2nd Media block)
	0	**Row Clock Gating Control Message** Access: R/W Gate Row Clocks Message : '0': Row Clock Un-gate Request (un-gates the crclk and cr2xclk clocks) '1': Row Clock Gate Request (gates the crclk and cr2xclk clocks)

CZWMRK - Color/Depth Write FIFO Watermarks

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04060h

This register is directly mapped to the current Virtual Addresses in the MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:24	**Reserved** Format: MBZ
	23:18	**Color Wr Burst Size** This is the maximum size of the requests burst, from the last High Watermark trip, before reevaluating the High Watermark again.
	17:16	**Reserved** Format: MBZ
	15:12	**Color Wr FIFO High Watermark** This is the number of accumulated Color writes that will trigger a Burst of Z Writes.
	11:6	**Z Wr Burst Size** This is the maximum size of the requests burst, from the last High Watermark trip, before reevaluating the High Watermark again.
	5:4	**Reserved** Format: MBZ
	3:0	**Z Wr FIFO High Watermark** This is the number of accumulated Depth writes that will trigger a Burst of Z Writes.

TS_GPGPU_THREADS_DISPATCHED - Count Active Channels Dispatched

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02290h

This register is used to count the number of active channels that TS sends for dispatch. For each dispatch the active bits in the execution mask are summed and added to this register. This register is reset when a write occurs to 2290h

DWord	Bit	Description
0	63:0	**GPGPU_THREADS_DISPATCHED** Format: U64 This count is increased by the number of active bits in the execution mask each time the TS sends a GPGPU dispatch.

CSPWRFSM - CS Power Management FSM

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	022ACh

This register contains the state code of the Power Management FSM, FBC Forward FSM, CSSTDT FSM, CSARB FSM, CSBUPDATE FSM. Decoding the contents of this register will indicate what the state of the corresponding state machine.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:30</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>29:28</td><td><b>CSFBCSLICE0</b>
<table><tr><td>Format:</td><td>U2</td></tr></table>
FBC message forward FSM state
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0h</td><td>CSFBCIDLE_0</td></tr>
<tr><td>1h</td><td>CSFBCMODIFY_0</td></tr>
<tr><td>2h</td><td>CSFBCCLEAN_0</td></tr>
<tr><td>3h</td><td>CSFBCDONE_0</td></tr>
</table></td></tr>
<tr><td></td><td>27:24</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>23:21</td><td><b>CS ARB</b>
<table><tr><td>Format:</td><td>U3</td></tr></table>
Overall state of the command streamer. Describes what state CS is in
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0h</td><td>ARBIDLE_s</td></tr>
<tr><td>1h</td><td>P0RNG_s CS</td></tr>
<tr><td>2h</td><td>P0BATCH_s</td></tr>
<tr><td>3h</td><td>ARBCHK</td></tr>
<tr><td>4h</td><td>ARBCHK1</td></tr>
<tr><td>5h</td><td>CTXOP_s</td></tr>
<tr><td>6h-7h</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td></td><td>20</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>19:17</td><td><b>CSSWITCH</b>
<table><tr><td>Format:</td><td>U3</td></tr></table></td></tr>
</table>

CSPWRFSM - CS Power Management FSM

Arbiters CSSWITCH FSM state decoding.

Value	Name
0h	SWIDLE_s
1h	SWITCH_s
2h	ASREQ_s
3h	DMACHK_s
4h	ARBWAIT_s
5h	FIFORECFG_s
6h-7h	Reserved

16:13 **CSCSBUPDATE**

Format:	U4

CS Power Management CSBLOCK FSM state

Value	Name
0h	CSBIDLE
1h	CSQ
2h	WRPTR
3h	SEMA1
4h	SEMA2
5h	TS1
6h	TS2
7h	TS3
8h	TS4
9h	DUMMYREQ
Ah	DUMMYWT
Bh	INTWT
Ch-Fh	Reserved

12:11 **R2MWRREQ**

Format:	U2

CSSTDT memory request FSM state

Value	Name
0h	WRIDLE
1h	WRREQ_HW1
2h	WRREQ_HW2
3h	WRRD

10 **Reserved**

Format:	MBZ

CSPWRFSM - CS Power Management FSM

9:7

LOADARB

Format:	U3

CSSTDT arbiter FSM state

Value	Name
0h	LDIDLE
1h	LDAUTO
2h	LDPRSR
3h	LDCTX
4h	LDFLSH
5h	LDREG
6h	LDSHR1
7h	Reserved

Programming Notes
LOADARB FSM states needs 4 bits for encoding, however only 3bits have been mapped on MMIO. 8h -> LDLRM is the state which is missed out, due to less bits mapped, LDLRM/LDIDLE cant be resolved with certain.

6:4

CSBLOCK

Format:	U3

CS Power Management CSBLOCK FSM state

Value	Name
0h	CSBLOCK
1h	CSCTXARB
2h	CSUNBLOCKRESTORE
3h	CSUNBLOCK
4h	CSPREP4BLOCK
5h-7h	Reserved

3:0

CSIDLE

Format:	U4

CS Power Management CSBLOCK FSM state

Value	Name
0h	CSBUSY
1h	CNTWT
2h	FLSHREQ
3h	FLSHWT
4h	CTXSAVE
5h	CSREQBLOCK

CSPWRFSM - CS Power Management FSM

		6h	PMTURNOFF
		7h	PMIDLEWT
		8h	IDLE
		9h	PMTURNON
		Ah	PMBUSYWT
		Bh	DOPFFCGREQ
		Ch	DOPFFCGWAIT
		Dh	DOPFFCG
		Eh	DOPFFCUGREQ
		Fh	DOPFFCUGWAIT

CTXTLB_VA - CTXTLB_VA Virtual page Address Registers

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24A00h-24A03h

This register is directly mapped to the current Virtual Addresses in the CTX TLB.

DWord	Bit	Description
0	31:12	**ADDRESS** Format: GraphicsAddress[31:12] PAGE VIRTUAL ADDRESS
	11:0	**RESERVED** Format: MBZ

CCID - Current Context Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02180h

This register contains the current logical rendering context address associated with the ring buffer in ring buffer mode of scheduling. This register contents are not valid in Exec-List mode of scheduling.

Programming Notes

The CCID register must not be written directly (via MMIO) unless the Command Streamer is completely idle (i.e., the Ring Buffer is empty and the pipeline is idle). Note that, under normal conditions, the CCID register should only be updated from the command stream using the MI_SET_CONTEXT command.

DWord	Bit	Description
0	31:12	**Logical Render Context Address (LRCA)** Default Value: 0h Format: GraphicsAddress[31:12] This field contains the 4 KB-aligned Graphics Memory Address of the current Logical Rendering Context. Bit 11 MBZ.
	11:10	**Reserved** Format: MBZ
	9	**Reserved**
	8	**Reserved** Format: Must Be One
	7:4	**Reserved** Format: MBZ
	3	**Extended State Save Enable** Format: Enable If set, the extended state identified in the Logical Context Data section of the Memory Data Formats chapter, is saved as part of switching away from this logical context.
	2	**Extended State Restore Enable** Format: Enable If set, the extended state identified in the Logical Context Data section of the Memory Data Formats chapter, was loaded (or restored) as part of switching to this logical context.
	1	**Reserved** Format: MBZ

CCID - Current Context Register

<table>
<tr>
<td></td>
<td>0</td>
<td><b>Valid</b>
<table>
<tr><td>Format:</td><td>U1</td></tr>
</table>
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td>Invalid <b>[Default]</b></td><td>The other fields of this register are invalid. A switch away from the context will not invoke a context save operation.</td></tr>
<tr><td>1h</td><td>Valid</td><td>The other fields of this register are valid, and a switch from the context will invoke the normal context save/restore operations.</td></tr>
</table>
</td>
</tr>
</table>

CEC0-0 - Customizable Event Creation 0-0

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02390h

This register is used to define custom counter event 0, bit definitions in this register refer to the CEC block diagram in the Custom Event Counters section.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:21</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>20:19</td><td><b>Source Select</b>
<table><tr><td>Format:</td><td>U2</td></tr></table>
Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>00b</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>01b</td><td>Prev Event</td><td>Selects the conditioned/flopped input from the last CEC block as the input bus to CEC0 block</td></tr>
<tr><td>10b</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>11b</td><td>Reserved</td><td></td></tr>
</table></td></tr>
<tr><td></td><td>18:3</td><td><b>Compare Value</b>
<table><tr><td>Format:</td><td>U16</td></tr></table>
<table><tr><td>The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.</td></tr></table></td></tr>
<tr><td></td><td>2:0</td><td><b>Compare Function</b>
<table><tr><td>Format:</td><td>U3</td></tr></table>
This field selects the function used by the CEC0 comparator when comparing the compare value to the value active on the CEC0 conditioned input bus (see block diagram in the Custom Event Counters section).
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>000b</td><td>Any Are Equal</td><td>Compare and assert output if any are equal (Can be used as OR function)</td></tr>
</table></td></tr>
</table>

CEC0-0 - Customizable Event Creation 0-0

		001b	Greater Than	Compare and assert output if greater than
		010b	Equal	Compare and assert output if equal to (Can also be used as AND function)
		011b	Greater Than or Equal	Compare and assert output if greater than or equal
		100b	Less Than	Compare and assert output if less than
		101b	Not Equal	Compare and assert output if not equal
		110b	Less Than or Equal	Compare and assert output if less than or equal
		111b	Reserved	

CEC1-0 - Customizable Event Creation 1-0

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02398h

This register is used to define custom counter event 1, bit definitions in this register refer to the CEC block diagram in the Custom Event Counters section.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:21</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table>
</td></tr>
<tr><td></td><td>20:19</td><td><b>Source Select</b>
<table><tr><td>Format:</td><td>U2</td></tr></table>
Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>00b</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>01b</td><td>Prev Event</td><td>Selects the conditioned/flopped input from the previous CEC block as the input bus to this CEC block</td></tr>
<tr><td>10b</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>11b</td><td>Reserved</td><td></td></tr>
</table>
</td></tr>
<tr><td></td><td>18:3</td><td><b>Compare Value</b>
<table><tr><td>Format:</td><td>U16</td></tr></table>
<table><tr><td>The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.</td></tr></table>
</td></tr>
<tr><td></td><td>2:0</td><td><b>Compare Function</b>
<table><tr><td>Format:</td><td>U3</td></tr></table>
This field selects the function used by the CEC comparator when comparing the compare value to the value active on the CEC conditioned input bus (see block diagram in the Custom Event Counters section).
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>000b</td><td>Any Are Equal</td><td>Compare and assert output if any are equal (Can be used as OR function)</td></tr>
<tr><td>001b</td><td>Greater Than</td><td>Compare and assert output if greater than</td></tr>
</table>
</td></tr>
</table>

CEC1-0 - Customizable Event Creation 1-0

		010b	Equal	Compare and assert output if equal to (Can also be used as AND function)
		011b	Greater Than or Equal	Compare and assert output if greater than or equal
		100b	Less Than	Compare and assert output if less than
		101b	Not Equal	Compare and assert output if not equal
		110b	Less Than or Equal	Compare and assert output if less than or equal
		111b	Reserved	

CEC2-0 - Customizable Event Creation 2-0

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	023A0h

This register is used to define custom counter event 2, bit definitions in this register refer to the CEC block diagram in the Custom Event Counters section.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:21</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>20:19</td><td><b>Source Select</b><table><tr><td>Format:</td><td>U2</td></tr></table>Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>00b</td><td>Reserved</td><td>Reserved</td></tr><tr><td>01b</td><td>Prev Event</td><td>Selects the conditioned/flopped input from the previous CEC block as the input bus to this CEC block</td></tr><tr><td>10b</td><td>Reserved</td><td>Reserved</td></tr><tr><td>11b</td><td>Reserved</td><td></td></tr></table></td></tr>
<tr><td></td><td>18:3</td><td><b>Compare Value</b><table><tr><td>Format:</td><td>U16</td></tr></table><table><tr><td>The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.</td></tr></table></td></tr>
<tr><td></td><td>2:0</td><td><b>Compare Function</b><table><tr><td>Format:</td><td>U3</td></tr></table>This field selects the function used by the CEC comparator when comparing the compare value to the value active on the CEC conditioned input bus (see block diagram in the Custom Event Counters section).<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>000b</td><td>Any Are Equal</td><td>Compare and assert output if any are equal (Can be used as OR function)</td></tr></table></td></tr>
</table>

CEC2-0 - Customizable Event Creation 2-0

		001b	Greater Than	Compare and assert output if greater than
		010b	Equal	Compare and assert output if equal to (Can also be used as AND function)
		011b	Greater Than or Equal	Compare and assert output if greater than or equal
		100b	Less Than	Compare and assert output if less than
		101b	Not Equal	Compare and assert output if not equal
		110b	Less Than or Equal	Compare and assert output if less than or equal
		111b	Reserved	

CEC3-0 - Customizable Event Creation 3-0

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	023A8h

This register is used to define custom counter event 3, bit definitions in this register refer to the CEC block diagram in the Custom Event Counters section.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:21</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table>
</td></tr>
<tr><td></td><td>20:19</td><td><b>Source Select</b>
<table><tr><td>Format:</td><td>U2</td></tr></table>
Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>00b</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>01b</td><td>Prev Event</td><td>Selects the conditioned/flopped input from the previous CEC block as the input bus to this CEC block</td></tr>
<tr><td>10b</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>11b</td><td>Reserved</td><td></td></tr>
</table>
</td></tr>
<tr><td></td><td>18:3</td><td><b>Compare Value</b>
<table><tr><td>Format:</td><td>U16</td></tr></table>
<table><tr><td>The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.</td></tr></table>
</td></tr>
<tr><td></td><td>2:0</td><td><b>Compare Function</b>
<table><tr><td>Format:</td><td>U3</td></tr></table>
This field selects the function used by the CEC comparator when comparing the compare value to the value active on the CEC conditioned input bus (see block diagram in the Custom Event Counters section).
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>000b</td><td>Any Are Equal</td><td>Compare and assert output if any are equal (Can be used as OR function)</td></tr>
</table>
</td></tr>
</table>

CEC3-0 - Customizable Event Creation 3-0

		001b	Greater Than	Compare and assert output if greater than
		010b	Equal	Compare and assert output if equal to (Can also be used as AND function)
		011b	Greater Than or Equal	Compare and assert output if greater than or equal
		100b	Less Than	Compare and assert output if less than
		101b	Not Equal	Compare and assert output if not equal
		110b	Less Than or Equal	Compare and assert output if less than or equal
		111b	Reserved	

CVS_TLB_LRA_0 - CVS TLB LRA 0

Register Space: MMIO: 0/2/0

Default Value: 0x1F080700

Size (in bits): 32

Address: 04044h-04047h

CVS TLB LRA 0

DWord	Bit	Description
0	31:29	**Reserved** Default Value: 000b Access: RO Reserved Format: MBZ
	28:24	**CVS LRA1 Max** Default Value: 11111b Access: R/W CVS LRA1 Max Format: MBZ Maximum value of programmable LRA1
	23:21	**Reserved** Default Value: 000b Access: RO Reserved Format: MBZ
	20:16	**CVS LRA1 Min** Default Value: 01000b Access: R/W CVS LRA1 Min Format: U6 Minimum value of programmable LRA1
	15:13	**Reserved** Default Value: 000b Access: RO Reserved

CVS_TLB_LRA_0 - CVS TLB LRA 0

		Format: MBZ
	12:8	**CVS LRA0 Max** Default Value: 00111b Access: R/W CVS LRA0 Max Format: MBZ Maximum value of programmable LRA0
	7:5	**Reserved** Default Value: 000b Access: RO Reserved Format: MBZ
	4:0	**CVS LRA0 Min** Default Value: 00000b Access: R/W CVS LRA0 Min Format: U6 Minimum value of programmable LRA0

CVS_TLB_LRA_1 - CVS TLB LRA 1

Register Space:	MMIO: 0/2/0
Default Value:	0x00001F18
Size (in bits):	32
Address:	04048h-0404Bh

CVS TLB LRA 1

DWord	Bit	Description
0	31:13	**Reserved** Default Value: 0000000000000000000b Access: RO Reserved Format: MBZ
	12:8	**CVS LRA2 Max** Default Value: 11111b Access: R/W CVS LRA2 Max Format: MBZ Maximum value of programmable LRA2
	7:5	**Reserved** Default Value: 000b Access: RO Reserved Format: MBZ
	4:0	**CVS LRA2 Min** Default Value: 11000b Access: R/W CVS LRA2 Min Format: U6 Minimum value of programmable LRA2

CVS_TLB_LRA_2 - CVS TLB LRA 2

Register Space:	MMIO: 0/2/0
Default Value:	0x00000005
Size (in bits):	32
Address:	0404Ch-0404Fh

CVS TLB LRA 2

DWord	Bit	Description
0	31:6	**Reserved** Default Value: 00000000000000000000000000b Access: RO Reserved Format: MBZ
	5:4	**CS LRA** Default Value: 00b Access: R/W CS LRA Format: U6 Which LRA should CS use
	3:2	**VF LRA** Default Value: 01b Access: R/W VF LRA Format: U1 Which LRA should VF use
	1:0	**SO LRA** Default Value: 01b Access: R/W SO LRA Format: MBZ Which LRA should SO use

ZSHR - Depth/Early Depth TLB Partitioning Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000020
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04050h

This register is used to determine the number of TLB entries from the total of 64 available to be used by the Depth partition of the TLB. The rest of the entries are used for the Early Depth/Stencil TLB.

DWord	Bit	Description
0	31:6	**Reserved** Format: MBZ
	5:0	**Number of TLB Entries Out of 64 used for Depth TLB** Default Value: 32 The rest are be used for Early Depth/Stencil TLB. Default value is 32.

DS_INVOCATION_COUNT - DS Invocation Counter

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02308h

This register stores the number of domain points shaded by the DS threads. Domain points which hit in the DS cache will not cause this register to increment. Note that the spawning of a DS thread which shades two domain points will cause this counter to increment by two. This register is part of the context save and restore.

DWord	Bit	Description
0	63:0	**DS Invocation Count** Number of domain points shaded by the DS threads. Updated only when DS Function Enable and Statistics Enable are set in 3DSTATE_DS

EIR - Error Identity Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W,RO
Size (in bits):	32
Address:	020B0h

The EIR register contains the persistent values of Hardware-Detected Error Condition bits. Any bit set in this register will cause the Master Error bit in the ISR to be set. The EIR register is also used by software to clear detected errors (by writing a 1 to the appropriate bit(s)), except for the unrecoverable bits described.)

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>15:0</td><td><b>Error Identity Bits</b><table><tr><td>Format:</td><td>Array of Error condition bits See the table titled Hardware-Detected Error Bits.</td></tr></table>This register contains the persistent values of ESR error status bits that are unmasked via the EMR register. (See Table Table 3-3. Hardware-Detected Error Bits). The logical OR of all (defined) bits in this register is reported in the Master Error bit of the Interrupt Status Register. In order to clear an error condition, software must first clear the error by writing a 1 to the appropriate bit(s) in this field. If required, software should then proceed to clear the Master Error bit of the IIR. Reserved bits are RO.<table><tr><th>Value</th><th>Name</th></tr><tr><td>1h</td><td>Error occurred</td></tr></table><table><tr><th>Programming Notes</th></tr><tr><td>Writing a 1 to a set bit will cause that error condition to be cleared. However, neither the Page Table Error bit (Bit 4) nor the Instruction Error bit (Bit 0) can be cleared except by reset (i.e., it is a fatal error).</td></tr></table></td></tr>
</table>

EMR - Error Mask Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x000000FF
Access:	R/W,RO
Size (in bits):	32
Address:	020B4h

The EMR register is used by software to control which Error Status Register bits are masked or unmasked. Unmasked bits will be reported in the EIR, thus setting the Master Error ISR bit and possibly triggering a CPU interrupt, and will persist in the EIR until cleared by software. Masked bits will not be reported in the EIR and therefore cannot generate Master Error conditions or CPU interrupts. Reserved bits are RO.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:8</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>Must Be One</td></tr></table>
<table><tr><th>Programming Notes</th></tr><tr><td>These bits are not implemented in HW and must be set to '1'</td></tr></table>
</td></tr>
<tr><td></td><td>7:0</td><td><b>Error Mask Bits</b>
<table><tr><td>Format:</td><td>Array of error condition mask bits See the table titled Hardware-Detected Error Bits.</td></tr></table>
This register contains a bit mask that selects which error condition bits (from the ESR) are reported in the EIR.
<table><tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>FFh</td><td><b>[Default]</b></td><td></td></tr>
<tr><td>0h</td><td>Not Masked</td><td>Will be reported in the EIR</td></tr>
<tr><td>1h</td><td>Masked</td><td>Will not be reported in the EIR</td></tr></table>
</td></tr>
</table>

ESR - Error Status Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	020B8h

The ESR register contains the current values of all Hardware-Detected Error condition bits (these are all by definition persistent). The EMR register selects which of these error conditions are reported in the persistent EIR (i.e., set bits must be cleared by software) and thereby causing a Master Error interrupt condition to be reported in the ISR.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>15:0</td><td><b>Error Status Bits</b>
<table><tr><td>Format:</td><td>Array of error condition bits See the table titled Hardware-Detected Error Bits.</td></tr></table>This register contains the non-persistent values of all hardware-detected error condition bits.<table><tr><th>Value</th><th>Name</th></tr><tr><td>1h</td><td>Error Condition Detected</td></tr></table></td></tr>
</table>

TD_PM_MODE_EUCOUNT - EU Mask Programming

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	WO
Size (in bits):	32

DWord	Bit	Description
0	31:24	**Reserved** Format: MBZ
	23	**SubSlice 2 EU 7 Enable** Format: Enable
	22	**SubSlice 2 EU 6 Enable** Format: Enable
	21	**SubSlice 2 EU 5 Enable** Format: Enable
	20	**SubSlice 2 EU 4 Enable** Format: Enable
	19	**SubSlice 2 EU 3 Enable** Format: Enable
	18	**SubSlice 2 EU 2 Enable** Format: Enable
	17	**SubSlice 2 EU 1 Enable** Format: Enable
	16	**SubSlice 2 EU 0 Enable** Format: Enable
	15	**SubSlice 1 EU 7 Enable** Format: Enable
	14	**SubSlice 1 EU 6 Enable** Format: Enable
	13	**SubSlice 1 EU 5 Enable** Format: Enable
	12	**SubSlice 1 EU 4 Enable** Format: Enable
	11	**SubSlice 1 EU 3 Enable** Format: Enable
	10	**SubSlice 1 EU 2 Enable**

TD_PM_MODE_EUCOUNT - EU Mask Programming

		Format: Enable
	9	**SubSlice 1 EU 1 Enable** Format: Enable
	8	**SubSlice 1 EU 0 Enable** Format: Enable
	7	**SubSlice 0 EU 7 Enable** Format: Enable
	6	**SubSlice 0 EU 6 Enable** Format: Enable
	5	**SubSlice 0 EU 5 Enable** Format: Enable
	4	**SubSlice 0 EU 4 Enable** Format: Enable
	3	**SubSlice 0 EU 3 Enable** Format: Enable
	2	**SubSlice 0 EU 2 Enable** Format: Enable
	1	**SubSlice 0 EU 1 Enable** Format: Enable
	0	**SubSlice 0 EU 0 Enable** Format: Enable

EXCC - Execute Condition Code Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W,RO
Size (in bits):	32
Trusted Type:	1
Address:	02028h

This register contains user defined and hardware generated conditions that are used by MI_WAIT_FOR_EVENT commands. An MI_WAIT_FOR_EVENT instruction excludes the executing ring from arbitration if the selected event evaluates to a 1, while instruction is discarded if the condition evaluates to a 0. Once excluded a ring is enabled into arbitration when the selected condition evaluates to a 0.
This register also contains control for the invalidation of indirect state pointers on context restore.

DWord	Bit	Description
0	31:16	**Mask Bits** Format: Mask[15:0] These bits serves as a write enable for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.
	15:12	**Reserved** Format: MBZ
	11	**Pending Indirect State Dirty Bit** Access: RO This field keeps track of whether or not an indirect state pointer command has been parsed in the current context. Clears either on a context save or explicitly through a flush command. This bit is Read Only.
	10:7	**Pending Indirect State Counter** This field keeps track of the maximum number of indirect state pointers pending in the system. When the register is saved/restored, it saves either a value of 1 or 0. This field is Read-Only.
	6:5	**Reserved** Format: MBZ
	4:0	**User Defined Condition Codes** The software may signal a Stream Semaphore by setting the Mask bit and Signal Bit together to match the bit field specified in a WAIT_FOR_EVENT (Semaphore).

GAB_ERR_REPORT - GAB Error Reporting Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24094h

This register is directly mapped for the Error Reporting Register.

DWord	Bit	Description
0	31:8	**Reserved**
	7	**HWSP GGTT fetch yields an invalid entry**
	6	**Reserved**
	5	**Reserved**
	4	**PD fetch yields an invalid entry**
	3	**PD fetch for entry marked as invalid by BCS**
	2	**GTT fetch yields an invalid entry** Page Fault occurred in one of the GTT translations.
	1	**Reserved**
	0	**CTXTLB fetch yields an invalid entry**

GAB_HWSP_REG - GAB Hardware Status Page Address Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04280h-04283h

This register is used to program the 4 KB-aligned System Memory address of the Hardware Status Page used to report hardware status into (typically cacheable) System Memory. This address in this register is translated using the Global GTT in memory. The mapping type of the GTT entry determines the snoop nature of the transaction to memory.

DWord	Bit	Description
0	31:12	**GAB HWSP Register** Default Value: 00000h Access: R/W
	11:0	**Reserved** Default Value: 000h Access: RO

GAB_CTL_REG - GAB unit Control Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x000000BF
Access:	R/W
Size (in bits):	32
Address:	24000h

DefaultValue=FF0000BFh Trusted Type = 1

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:9</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>8</td><td><b>Continue after Page Fault</b>
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>1</td><td>GAB Set</td><td>Ipon receiving a page fault when requesting an address translation, GAB will set address bit 39 to 1 and continue.</td></tr>
<tr><td>0</td><td>GAB Hang</td><td>GAB will hang on a page fault. Default = b0.</td></tr>
</table></td></tr>
<tr><td></td><td>7:6</td><td><b>PPGTT BCS TLB LRA MIN</b>
<table><tr><td>Default Value:</td><td>10b</td></tr></table>
TLB Depth Partitioning Register In PP GTT Mode.</td></tr>
<tr><td></td><td>5:4</td><td><b>GAB write request priority signal value used in GAC arbitration</b>
<table><tr><td>Default Value:</td><td>11b</td></tr></table></td></tr>
<tr><td></td><td>3:2</td><td><b>GAB read only request priority signal value used in GAC arbitration</b>
<table><tr><td>Default Value:</td><td>11b</td></tr></table></td></tr>
<tr><td></td><td>1:0</td><td><b>GAB read request priority signal value used in GAC arbitration</b>
<table><tr><td>Default Value:</td><td>11b</td></tr></table></td></tr>
</table>

GAC_ARB_CTL_REG - GAC_GAB Arbitration Counters Register 1

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00400002
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	14050h

GAC_GAB R/RO/W Arbitration Control Register

DWord	Bit	Description
0	31	**GAC write request Limit Enable** Format: U1 As long As there is no conflict between GAC and GAB ,GAC will alow whoever shows up (if media present and no GAB, let meda and vice versa). If both are present, start counting and switch when programmable no of request is expired. Allow only One GAB request and reset the counter. Counter only counts while we service a particular client and another client is present, else counter will reset.
	30	**VLF Final write Limit Enable** Format: MBZ As long as there is no conflict Between VCS MFD and VLF Final Write, GAC will allow whoever shows up (if VLF present and no VCSMFD, Let VLF and vice versa). If both are present, Start counting and when programmable no of request is expired. Allow only One VCSMFD request And counter will reset. Counter only counts while we service a particular client and another client is present, else counter will reset.
	29:24	**Write Req Limit Count** Format: U6 The value programmed determines the number of GAC/VLF Writes will allow for Each time.
	23	**GAC/GAB Cascaded Read Only Limit Enable** Format: U1 As long as there is no conflict between GAC and GAB Read Requests, GAC will allow whoever shows up (if GAC present and no GAB, Let GAC and vice versa). If both are present, Start counting and switch when programmable no of request from either side is expired (reset the counter when switch). Counter only counts while we service a particular client and other client is present, else counter will reset.
	22	**Fixed Priority Setting** Format: MBZ

GAC_ARB_CTL_REG - GAC_GAB Arbitration Counters Register 1

<table>
<tr><td></td><td></td><td>Once programmable counter is disabled, GAC uses the fixed arbitration setting given in this register setting.
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0</td><td>GAC</td></tr>
<tr><td>1</td><td>GAB <b>[Default]</b></td></tr>
</table></td></tr>
<tr><td></td><td>21</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>20:16</td><td><b>GAC/GAB Read Only Limit Counter Value</b>
<table><tr><td>Format:</td><td>U5</td></tr></table>
This is the Maximum number of Read requests Allowed from Each Cascaded Agent. Default 0</td></tr>
<tr><td></td><td>15</td><td><b>GAC/GAB Cascaded Read Limit Enable</b>
<table><tr><td>Format:</td><td>U1</td></tr></table>
<table>
<tr><td>As long as there is no conflict Between GAC and GAB Read Only Requests, GAC will allow whoever shows up (if GAC present and no GAB, Let GAC and vice versa). If both are present, Start counting and switch when programmable no of request from either side is expired (reset the counter when switch). Counter only counts while we service a particular client and other client is present, else counter will reset.</td></tr>
<tr><td>Default 0</td></tr>
</table></td></tr>
<tr><td></td><td>14</td><td><b>Default priority 0-GAC, 1-GAB</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table>
Default 0</td></tr>
<tr><td></td><td>13</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>12:8</td><td><b>GAC/GAB Read Limit Counter Value</b>
<table><tr><td>Format:</td><td>U5</td></tr></table>
This is the Maximum number of Read requests allowed from Each Cascaded Agent.</td></tr>
<tr><td></td><td>7:6</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>5:0</td><td><b>No of Global GTT Entries Valid in PPGTT mode in TLB064</b>
<table>
<tr><td>Default Value:</td><td>000010b</td></tr>
<tr><td>Format:</td><td>U6</td></tr>
</table>
Minimum value the PPGGTT LRA can have (effectively partitioning the TLB between PPGTT and GGTT). Currently, only 2 entries are allocated to GGTT in ASmodel. TLB64 is shared by GGTT and PPGTT entries, are 2 LRAs, the GGTT one running from 0 up to PPGTT_MIN -1 (which is 2, but</td></tr>
</table>

GAC_ARB_CTL_REG - GAC_GAB Arbitration Counters Register 1

		could be changed if needed), and the PPGTT one running from PPGTT_MIN up to 63.

ARB_GAC_GAM_REQCNTS0 - GAC_GAM Arbitration Counters Register 0

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043A8h

DWord	Bit	Description
0	31:22	**Reserved**
	21:16	**Number of GAC WR requests to be accumulated before applying the arbitration**
	15:14	**Reserved**
	13:8	**Number of GAC R requests to be accumulated before applying the arbitration**
	7:6	**Reserved**
	5:0	**Number of GAC RO requests to be accumulated before applying the arbitration**

ARB_GAC_GAM_REQCNTS1 - GAC_GAM Arbitration Counters Register 1

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043ACh

DWord	Bit	Description
0	31:22	**Reserved**
	21:16	**Number of GAC WR requests to be accumulated before applying the arbitration**
	15:14	**Reserved**
	13:8	**Number of GAC R requests to be accumulated before applying the arbitration**
	7:6	**Reserved**
	5:0	**Number of GAC RO requests to be accumulated before applying the arbitration**

ARB_R_GAC_GAM0 - GAC_GAM R Arbitration Register 0

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043E0h

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 1**
	26:24	**Goto field for entry 1 when request vector is 11b**
	23:21	**Goto field for entry 1 when request vector is 10b**
	20:18	**Goto field for entry 1 when request vector is 01b**
	17:15	**Goto field for entry 1 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 0**
	11:9	**Goto field for entry 0 when request vector is 11b**
	8:6	**Goto field for entry 0 when request vector is 10b**
	5:3	**Goto field for entry 0 when request vector is 01b**
	2:0	**Goto field for entry 0 when request vector is 00b**

ARB_R_GAC_GAM1 - GAC_GAM R Arbitration Register 1

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043E4h

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 3**
	26:24	**Goto field for entry 3 when request vector is 11b**
	23:21	**Goto field for entry 3 when request vector is 10b**
	20:18	**Goto field for entry 3 when request vector is 01b**
	17:15	**Goto field for entry 3 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 2**
	11:9	**Goto field for entry 2 when request vector is 11b**
	8:6	**Goto field for entry 2 when request vector is 10b**
	5:3	**Goto field for entry 2 when request vector is 01b**
	2:0	**Goto field for entry 2 when request vector is 00b**

ARB_R_GAC_GAM2 - GAC_GAM R Arbitration Register 2

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043E8h

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 5**
	26:24	**Goto field for entry 5 when request vector is 11b**
	23:21	**Goto field for entry 5 when request vector is 10b**
	20:18	**Goto field for entry 5 when request vector is 01b**
	17:15	**Goto field for entry 5 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 4**
	11:9	**Goto field for entry 4 when request vector is 11b**
	8:6	**Goto field for entry 4 when request vector is 10b**
	5:3	**Goto field for entry 4 when request vector is 01b**
	2:0	**Goto field for entry 4 when request vector is 00b**

ARB_R_GAC_GAM3 - GAC_GAM R Arbitration Register 3

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043ECh

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 7**
	26:24	**Goto field for entry 7 when request vector is 11b**
	23:21	**Goto field for entry 7 when request vector is 10b**
	20:18	**Goto field for entry 7 when request vector is 01b**
	17:15	**Goto field for entry 7 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 6**
	11:9	**Goto field for entry 6 when request vector is 11b**
	8:6	**Goto field for entry 6 when request vector is 10b**
	5:3	**Goto field for entry 6 when request vector is 01b**
	2:0	**Goto field for entry 6 when request vector is 00b**

ARB_RO_GAC_GAM0 - GAC_GAM RO Arbitration Register 0

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043D0h

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 1**
	26:24	**Goto field for entry 1 when request vector is 11b**
	23:21	**Goto field for entry 1 when request vector is 10b**
	20:18	**Goto field for entry 1 when request vector is 01b**
	17:15	**Goto field for entry 1 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 01**
	11:9	**Goto field for entry 01 when request vector is 11b**
	8:6	**Goto field for entry 01 when request vector is 10b**
	5:3	**Goto field for entry 01 when request vector is 01b**
	2:0	**Goto field for entry 01 when request vector is 00b**

ARB_RO_GAC_GAM1 - GAC_GAM RO Arbitration Register 1

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043D4h

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 3**
	26:24	**Goto field for entry 3 when request vector is 11b**
	23:21	**Goto field for entry 3 when request vector is 10b**
	20:18	**Goto field for entry 3 when request vector is 01b**
	17:15	**Goto field for entry 3 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 2**
	11:9	**Goto field for entry 2 when request vector is 11b**
	8:6	**Goto field for entry 2 when request vector is 10b**
	5:3	**Goto field for entry 2 when request vector is 01b**
	2:0	**Goto field for entry 2 when request vector is 00b**

ARB_RO_GAC_GAM2 - GAC_GAM RO Arbitration Register 2

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043D8h

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 5**
	26:24	**Goto field for entry 5 when request vector is 11b**
	23:21	**Goto field for entry 5 when request vector is 10b**
	20:18	**Goto field for entry 5 when request vector is 01b**
	17:15	**Goto field for entry 5 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 4**
	11:9	**Goto field for entry 4 when request vector is 11b**
	8:6	**Goto field for entry 4 when request vector is 10b**
	5:3	**Goto field for entry 4 when request vector is 01b**
	2:0	**Goto field for entry 4 when request vector is 00b**

ARB_RO_GAC_GAM3 - GAC_GAM RO Arbitration Register 3

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043DCh

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 7**
	26:24	**Goto field for entry 7 when request vector is 11b**
	23:21	**Goto field for entry 7 when request vector is 10b**
	20:18	**Goto field for entry 7 when request vector is 01b**
	17:15	**Goto field for entry 7 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 6**
	11:9	**Goto field for entry 6 when request vector is 11b**
	8:6	**Goto field for entry 6 when request vector is 10b**
	5:3	**Goto field for entry 6 when request vector is 01b**
	2:0	**Goto field for entry 6 when request vector is 00b**

ARB_WR_GAC_GAM0 - GAC_GAM WR Arbitration Register 0

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043F0h

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 1**
	26:24	**Goto field for entry 1 when request vector is 11b**
	23:21	**Goto field for entry 1 when request vector is 10b**
	20:18	**Goto field for entry 1 when request vector is 01b**
	17:15	**Goto field for entry 1 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 0**
	11:9	**Goto field for entry 0 when request vector is 11b**
	8:6	**Goto field for entry 0 when request vector is 10b**
	5:3	**Goto field for entry 0 when request vector is 01b**
	2:0	**Goto field for entry 0 when request vector is 00b**

ARB_WR_GAC_GAM1 - GAC_GAM WR Arbitration Register 1

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043F4h

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 3**
	26:24	**Goto field for entry 3 when request vector is 11b**
	23:21	**Goto field for entry 3 when request vector is 10b**
	20:18	**Goto field for entry 3 when request vector is 01b**
	17:15	**Goto field for entry 3 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 2**
	11:9	**Goto field for entry 2 when request vector is 11b**
	8:6	**Goto field for entry 2 when request vector is 10b**
	5:3	**Goto field for entry 2 when request vector is 01b**
	2:0	**Goto field for entry 2 when request vector is 00b**

ARB_WR_GAC_GAM2 - GAC_GAM WR Arbitration Register 2

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043F8h

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 5**
	26:24	**Goto field for entry 5 when request vector is 11b**
	23:21	**Goto field for entry 5 when request vector is 10b**
	20:18	**Goto field for entry 5 when request vector is 01b**
	17:15	**Goto field for entry 5 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 4**
	11:9	**Goto field for entry 4 when request vector is 11b**
	8:6	**Goto field for entry 4 when request vector is 10b**
	5:3	**Goto field for entry 4 when request vector is 01b**
	2:0	**Goto field for entry 4 when request vector is 00b**

ARB_WR_GAC_GAM3 - GAC_GAM WR Arbitration Register 3

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043FCh

DWord	Bit	Description
0	31:28	**Reserved**
	27	**Priority for entry 7**
	26:24	**Goto field for entry 7 when request vector is 11b**
	23:21	**Goto field for entry 7 when request vector is 10b**
	20:18	**Goto field for entry 7 when request vector is 01b**
	17:15	**Goto field for entry 7 when request vector is 00b**
	14:13	**Reserved**
	12	**Priority for entry 6**
	11:9	**Goto field for entry 6 when request vector is 11b**
	8:6	**Goto field for entry 6 when request vector is 10b**
	5:3	**Goto field for entry 6 when request vector is 01b**
	2:0	**Goto field for entry 6 when request vector is 00b**

GAC_HWSP_REG - GAC Hardware Status Page Address Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04180h-04183h

This register is used to program the 4 KB-aligned System Memory address of the Hardware Status Page used to report hardware status into (typically cacheable) System Memory. This address in this register is translated using the Global GTT in memory. The mapping type of the GTT entry determines the snoop nature of the transaction to memory.

DWord	Bit	Description
0	31:12	**GAC HWSP Register** Default Value: 00000h Access: R/W
	11:0	**Reserved** Default Value: 000h Access: RO

DONE_REG - GAM Fub Done Lookup Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	040B0h-040B3h

Gam Fub Done Lookup Register

DWord	Bit	Description
0	31:0	**Gam Fub Done Lookup Reg** Default Value: 00000000h Access: RO 31 CVS Credit Fifo is Empty 30 CVS TLB Don't have any Cycles 29 Z Credit fifo is empty 28 ZTLB Don't have any cycles 27 RCC Credit Fifo is empty 26 RCC TLB Don't have any cycles 25 L3 Credit fifo is empty 24 L3 TLB is don't have any Cycles 23 VLF Credit fifo is empty 22 VLF TLB don't have any cycles 21 CASC Credit fifo empty 20 CASC TLB don't have any Cycles 19 Miss Fub Done 18 Read Stream Done 17 Read Steam Fifo is empty 16 Recycle Fifo in rstrm is empty

DONE_REG - GAM Fub Done Lookup Register

		15	TLB Pend Done
		14	TLB Pend PQ Array Is done
		13	TLB pend PB Array is done
		12	Read route fub is done
		11	Gafm Data fifo is empty
		10	GAP data fifo is empty
		9	GAC data fifo is empty
		8	Wrdp is done with all the cycles
		7	Wrdp RID fifo is empty
		6	No hold from midarb to RTSTRM
		5	No hold from TLBPEND to MIDARB
		4	Reserved
		3	Tied to "1" - to be defined
		2	Fence FSM are IDLE
		1	Non PD Load Done
		0	Tied to "1" - to be defined

GAM_HWSP_REG - GAM Hardware Status Page Address Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04080h-04083h

This register is used to program the 4 KB-aligned System Memory address of the Hardware Status Page used to report hardware status into (typically cacheable) System Memory. This address in this register is translated using the Global GTT in memory. The mapping type of the GTT entry determines the snoop nature of the transaction to memory.

DWord	Bit	Description
0	31:12	**GAM HWSP Register** Default Value: 00000h Access: R/W
	11:0	**Reserved** Default Value: 000h Access: RO

GFX_PRIO_CTRL - GFX Arbiter Client Priority Control

Register Space:	MMIO: 0/2/0
Default Value:	0x00011D10
Size (in bits):	32
Address:	0402Ch-0402Fh

GFX Arbiter Client Priority Control

DWord	Bit	Description
0	31:17	**Extra 402C Register** Default Value: 000000000000000b Access: R/W Extra 402C Register
	16:12	**Read Rstrm Max Reject** Default Value: 10001b Access: R/W Read Rstrm Max Reject
	11:9	**gapc_gam_c_priority** Default Value: 110b Access: R/W gapc_gam_c_priority - Lowest Bit [9] is not used
	8:6	**gapc_gam_z_priority** Default Value: 100b Access: R/W gapc_gam_z_priority - Lowest Bit [6] is not used
	5:3	**gapc_gam_l3_priority** Default Value: 010b Access: R/W gapc_gam_l3_priority - Lowest Bit [3] is not used

GFX_PRIO_CTRL - GFX Arbiter Client Priority Control

	2:0	**gafm_gam_priority** Default Value: 000b Access: R/W Client Priority control bitss gafm_gam_priority - Lowest Bit [0] is not used

MIDARB_GOTOFIELD_HIT0 - Goto Field in Programmable Arbitration for Hit0

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043B0h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>15:14</td><td><b>Goto field when request vector is 111</b>
Determines the GOTO and priority register to be used next:<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>00b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT0 and MIDARB_PRIO_HIT_REGISTER[2:0]</td></tr><tr><td>01b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT1 and MIDARB_PRIO_HIT_REGISTER[5:3]</td></tr><tr><td>10b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT2 and MIDARB_PRIO_HIT_REGISTER[8:6]</td></tr><tr><td>11b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT3 and MIDARB_PRIO_HIT_REGISTER[11:9]</td></tr></table></td></tr>
<tr><td></td><td>13:12</td><td><b>Goto field when request vector is 110b.</b></td></tr>
<tr><td></td><td>11:10</td><td><b>Goto field when request vector is 101b.</b></td></tr>
<tr><td></td><td>9:8</td><td><b>Goto field when request vector is 100b.</b></td></tr>
<tr><td></td><td>7:6</td><td><b>Goto field when request vector is 011b.</b></td></tr>
<tr><td></td><td>5:4</td><td><b>Goto field when request vector is 010b.</b></td></tr>
<tr><td></td><td>3:2</td><td><b>Goto field when request vector is 001b.</b></td></tr>
<tr><td></td><td>1:0</td><td><b>Goto field when request vector is 000b.</b></td></tr>
</table>

MIDARB_GOTOFIELD_HIT1 - Goto Field in Programmable Arbitration for Hit1

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043B4h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>15:14</td><td><b>Goto field when request vector is 111</b>
Determines the GOTO and priority register to be used next<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>00b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT0 and MIDARB_PRIO_HIT_REGISTER[2:0]</td></tr><tr><td>01b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT1 and MIDARB_PRIO_HIT_REGISTER[5:3]</td></tr><tr><td>10b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT2 and MIDARB_PRIO_HIT_REGISTER[8:6]</td></tr><tr><td>11b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT3 and MIDARB_PRIO_HIT_REGISTER[11:9]</td></tr></table></td></tr>
<tr><td></td><td>13:12</td><td><b>Goto field when request vector is 110b.</b></td></tr>
<tr><td></td><td>11:10</td><td><b>Goto field when request vector is 101b.</b></td></tr>
<tr><td></td><td>9:8</td><td><b>Goto field when request vector is 100b.</b></td></tr>
<tr><td></td><td>7:6</td><td><b>Goto field when request vector is 011b.</b></td></tr>
<tr><td></td><td>5:4</td><td><b>Goto field when request vector is 010b.</b></td></tr>
<tr><td></td><td>3:2</td><td><b>Goto field when request vector is 001b.</b></td></tr>
<tr><td></td><td>1:0</td><td><b>Goto field when request vector is 000b.</b></td></tr>
</table>

MIDARB_GOTOFIELD_HIT2 - Goto Field in Programmable Arbitration for Hit2

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043B8h

DWord	Bit	Description
0	31:16	**Reserved** Format: MBZ
	15:14	**Goto field when request vector is 111.** Determines the GOTO and priority register to be used next
	13:12	**Goto field when request vector is 110b.**
	11:10	**Goto field when request vector is 101b.**
	9:8	**Goto field when request vector is 100b.**
	7:6	**Goto field when request vector is 011b.**
	5:4	**Goto field when request vector is 010b.**
	3:2	**Goto field when request vector is 001b.**
	1:0	**Goto field when request vector is 000b.**

Bits 15:14 values:

Value	Name	Description
00b		Use MIDARB_GOTOFIELD_HIT0 and MIDARB_PRIO_HIT_REGISTER[2:0]
01b		Use MIDARB_GOTOFIELD_HIT1 and MIDARB_PRIO_HIT_REGISTER[5:3]
10b		Use MIDARB_GOTOFIELD_HIT2 and MIDARB_PRIO_HIT_REGISTER[8:6]
11b		Use MIDARB_GOTOFIELD_HIT3 and MIDARB_PRIO_HIT_REGISTER[11:9]

MIDARB_GOTOFIELD_HIT3 - Goto Field in Programmable Arbitration for Hit3

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043BCh

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>15:14</td><td><b>Goto field when request vector is 111.</b>
Determines the GOTO and priority register to be used next.
Field for arbitration on next clock cycle for request entries of 111 corresponding to arbitration action field entry of MIDARB_PRIO_HIT_REGISTER[11:9]<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>00b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT0 and MIDARB_PRIO_HIT_REGISTER[2:0]</td></tr><tr><td>01b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT1 and MIDARB_PRIO_HIT_REGISTER[5:3]</td></tr><tr><td>10b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT2 and MIDARB_PRIO_HIT_REGISTER[8:6]</td></tr><tr><td>11b</td><td></td><td>Use MIDARB_GOTOFIELD_HIT3 and MIDARB_PRIO_HIT_REGISTER[11:9]</td></tr></table></td></tr>
<tr><td></td><td>13:12</td><td><b>Goto field when request vector is 110.</b>
Field for arbitration on next clock cycle for request entries of 110 corresponding to arbitration action field entry of MIDARB_PRIO_HIT_REGISTER[11:9]</td></tr>
<tr><td></td><td>11:10</td><td><b>Goto field when request vector is 101.</b>
Field for arbitration on next clock cycle for request entries of 101 corresponding to arbitration action field entry of MIDARB_PRIO_HIT_REGISTER[11:9]</td></tr>
<tr><td></td><td>9:8</td><td><b>Goto field when request vector is 100.</b>
Field for arbitration on next clock cycle for request entries of 100 corresponding to arbitration action field entry of MIDARB_PRIO_HIT_REGISTER[11:9]</td></tr>
<tr><td></td><td>7:6</td><td><b>Goto field when request vector is 011.</b>
Field for arbitration on next clock cycle for request entries of 011 corresponding to arbitration action field entry of MIDARB_PRIO_HIT_REGISTER[11:9]</td></tr>
<tr><td></td><td>5:4</td><td><b>Goto field when request vector is 010.</b>
Field for arbitration on next clock cycle for request entries of 010 corresponding to arbitration action field entry of MIDARB_PRIO_HIT_REGISTER[11:9]</td></tr>
<tr><td></td><td>3:2</td><td><b>Goto field when request vector is 001.</b>
Field for arbitration on next clock cycle for request entries of 001 corresponding to arbitration action field entry of MIDARB_PRIO_HIT_REGISTER[11:9]</td></tr>
<tr><td></td><td>1:0</td><td><b>Goto field when request vector is 000.</b></td></tr>
</table>

MIDARB_GOTOFIELD_HIT3 - Goto Field in Programmable Arbitration for Hit3

		Field for arbitration on next clock cycle for request entries of 000 corresponding to arbitration action field entry of MIDARB_PRIO_HIT_REGISTER[11:9]

MIDARB_GOTOFIELD_NP0 - Goto Field in Programmable Arbitration for Hit-NP0

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043C0h

DWord	Bit	Description
0	31:30	**Goto field when request vector is 1111.** Determines the GOTO and priority register to be used next. <table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>00b</td><td></td><td>Use MIDARB_GOTOFIELD_NP0 and MIDARB_PRIO_NP_REGISTER[4:0]</td></tr><tr><td>01b</td><td></td><td>Use MIDARB_GOTOFIELD_NP1 and MIDARB_PRIO_NP_REGISTER[9:5]</td></tr><tr><td>10b</td><td></td><td>Use MIDARB_GOTOFIELD_NP2 and MIDARB_PRIO_NP_REGISTER[14:10]</td></tr><tr><td>11b</td><td></td><td>Use MIDARB_GOTOFIELD_NP3 and MIDARB_PRIO_NP_REGISTER[19:15]</td></tr></table>
	29:28	**Goto field when request vector is 1110b.**
	27:26	**Goto field when request vector is 1101b.**
	25:24	**Goto field when request vector is 1100b.**
	23:22	**Goto field when request vector is 1011b.**
	21:20	**Goto field when request vector is 1010b.**
	19:18	**Goto field when request vector is 1001b.**
	17:16	**Goto field when request vector is 1000b.**
	15:14	**Goto field when request vector is 0111b.**
	13:12	**Goto field when request vector is 0110b.**
	11:10	**Goto field when request vector is 0101b.**
	9:8	**Goto field when request vector is 0100b.**
	7:6	**Goto field when request vector is 0011b.**
	5:4	**Goto field when request vector is 0010b.**
	3:2	**Goto field when request vector is 0001b.**
	1:0	**Goto field when request vector is 0000b.**

MIDARB_GOTOFIELD_NP1 - Goto Field in Programmable Arbitration for Hit-NP1

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043C4h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:30</td><td><b>Goto field when request vector is 1111.</b>
Determines the GOTO and priority register to be used next.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>00b</td><td></td><td>Use MIDARB_GOTOFIELD_NP0 and MIDARB_PRIO_NP_REGISTER[4:0]</td></tr>
<tr><td>01b</td><td></td><td>Use MIDARB_GOTOFIELD_NP1 and MIDARB_PRIO_NP_REGISTER[9:5]</td></tr>
<tr><td>10b</td><td></td><td>Use MIDARB_GOTOFIELD_NP2 and MIDARB_PRIO_NP_REGISTER[14:10]</td></tr>
<tr><td>11b</td><td></td><td>Use MIDARB_GOTOFIELD_NP3 and MIDARB_PRIO_NP_REGISTER[19:15]</td></tr>
</table></td></tr>
<tr><td></td><td>29:28</td><td><b>Goto field when request vector is 1110b.</b></td></tr>
<tr><td></td><td>27:26</td><td><b>Goto field when request vector is 1101b.</b></td></tr>
<tr><td></td><td>25:24</td><td><b>Goto field when request vector is 1100b.</b></td></tr>
<tr><td></td><td>23:22</td><td><b>Goto field when request vector is 1011b.</b></td></tr>
<tr><td></td><td>21:20</td><td><b>Goto field when request vector is 1010b.</b></td></tr>
<tr><td></td><td>19:18</td><td><b>Goto field when request vector is 1001b.</b></td></tr>
<tr><td></td><td>17:16</td><td><b>Goto field when request vector is 1000b.</b></td></tr>
<tr><td></td><td>15:14</td><td><b>Goto field when request vector is 0111b.</b></td></tr>
<tr><td></td><td>13:12</td><td><b>Goto field when request vector is 0110b.</b></td></tr>
<tr><td></td><td>11:10</td><td><b>Goto field when request vector is 0101b.</b></td></tr>
<tr><td></td><td>9:8</td><td><b>Goto field when request vector is 0100b.</b></td></tr>
<tr><td></td><td>7:6</td><td><b>Goto field when request vector is 0011b.</b></td></tr>
<tr><td></td><td>5:4</td><td><b>Goto field when request vector is 0010b.</b></td></tr>
<tr><td></td><td>3:2</td><td><b>Goto field when request vector is 0001b.</b></td></tr>
<tr><td></td><td>1:0</td><td><b>Goto field when request vector is 0000b.</b></td></tr>
</table>

MIDARB_GOTOFIELD_NP2 - Goto Field in Programmable Arbitration for Hit-NP2

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043C8h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:30</td><td><b>Goto field when request vector is 1111.</b>
Determines the GOTO and priority register to be used next.
<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>00b</td><td></td><td>Use MIDARB_GOTOFIELD_NP0 and MIDARB_PRIO_NP_REGISTER[4:0]</td></tr><tr><td>01b</td><td></td><td>Use MIDARB_GOTOFIELD_NP1 and MIDARB_PRIO_NP_REGISTER[9:5]</td></tr><tr><td>10b</td><td></td><td>Use MIDARB_GOTOFIELD_NP2 and MIDARB_PRIO_NP_REGISTER[14:10]</td></tr><tr><td>11b</td><td></td><td>Use MIDARB_GOTOFIELD_NP3 and MIDARB_PRIO_NP_REGISTER[19:15]</td></tr></table></td></tr>
<tr><td></td><td>29:28</td><td><b>Goto field when request vector is 1110b.</b></td></tr>
<tr><td></td><td>27:26</td><td><b>Goto field when request vector is 1101b.</b></td></tr>
<tr><td></td><td>25:24</td><td><b>Goto field when request vector is 1100b.</b></td></tr>
<tr><td></td><td>23:22</td><td><b>Goto field when request vector is 1011b.</b></td></tr>
<tr><td></td><td>21:20</td><td><b>Goto field when request vector is 1010b.</b></td></tr>
<tr><td></td><td>19:18</td><td><b>Goto field when request vector is 1001b.</b></td></tr>
<tr><td></td><td>17:16</td><td><b>Goto field when request vector is 1000b.</b></td></tr>
<tr><td></td><td>15:14</td><td><b>Goto field when request vector is 0111b.</b></td></tr>
<tr><td></td><td>13:12</td><td><b>Goto field when request vector is 0110b.</b></td></tr>
<tr><td></td><td>11:10</td><td><b>Goto field when request vector is 0101b.</b></td></tr>
<tr><td></td><td>9:8</td><td><b>Goto field when request vector is 0100b.</b></td></tr>
<tr><td></td><td>7:6</td><td><b>Goto field when request vector is 0011b.</b></td></tr>
<tr><td></td><td>5:4</td><td><b>Goto field when request vector is 0010b.</b></td></tr>
<tr><td></td><td>3:2</td><td><b>Goto field when request vector is 0001b.</b></td></tr>
<tr><td></td><td>1:0</td><td><b>Goto field when request vector is 0000b.</b></td></tr>
</table>

MIDARB_GOTOFIELD_NP3 - Goto Field in Programmable Arbitration for Hit-NP3

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043CCh

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:30</td><td><b>Goto field when request vector is 1111.</b>
Determines the GOTO and priority register to be used next.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>00b</td><td></td><td>Use MIDARB_GOTOFIELD_NP0 and MIDARB_PRIO_NP_REGISTER[4:0]</td></tr>
<tr><td>01b</td><td></td><td>Use MIDARB_GOTOFIELD_NP1 and MIDARB_PRIO_NP_REGISTER[9:5]</td></tr>
<tr><td>10b</td><td></td><td>Use MIDARB_GOTOFIELD_NP2 and MIDARB_PRIO_NP_REGISTER[14:10]</td></tr>
<tr><td>11b</td><td></td><td>Use MIDARB_GOTOFIELD_NP3 and MIDARB_PRIO_NP_REGISTER[19:15]</td></tr>
</table></td></tr>
<tr><td></td><td>29:28</td><td><b>Goto field when request vector is 1110b.</b></td></tr>
<tr><td></td><td>27:26</td><td><b>Goto field when request vector is 1101b.</b></td></tr>
<tr><td></td><td>25:24</td><td><b>Goto field when request vector is 1100b.</b></td></tr>
<tr><td></td><td>23:22</td><td><b>Goto field when request vector is 1011b.</b></td></tr>
<tr><td></td><td>21:20</td><td><b>Goto field when request vector is 1010b.</b></td></tr>
<tr><td></td><td>19:18</td><td><b>Goto field when request vector is 1001b.</b></td></tr>
<tr><td></td><td>17:16</td><td><b>Goto field when request vector is 1000b.</b></td></tr>
<tr><td></td><td>15:14</td><td><b>Goto field when request vector is 0111b.</b></td></tr>
<tr><td></td><td>13:12</td><td><b>Goto field when request vector is 0110b.</b></td></tr>
<tr><td></td><td>11:10</td><td><b>Goto field when request vector is 0101b.</b></td></tr>
<tr><td></td><td>9:8</td><td><b>Goto field when request vector is 0100b.</b></td></tr>
<tr><td></td><td>7:6</td><td><b>Goto field when request vector is 0011b.</b></td></tr>
<tr><td></td><td>5:4</td><td><b>Goto field when request vector is 0010b.</b></td></tr>
<tr><td></td><td>3:2</td><td><b>Goto field when request vector is 0001b.</b></td></tr>
<tr><td></td><td>1:0</td><td><b>Goto field when request vector is 0000b.</b></td></tr>
</table>

GPGPU_DISPATCHDIMX - GPGPU Dispatch Dimension X

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02500h

DWord	Bit	Description
0	31:0	**Dispatch Dimension X** Format: U32 The number of thread groups to be dispatched in the X dimension (max x + 1). Value: 1,FFFFFFFFh / Name:

GPGPU_DISPATCHDIMY - GPGPU Dispatch Dimension Y

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02504h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:0</td><td><b>Dispatch Dimension Y</b>
<table><tr><td>Format:</td><td>U32</td></tr></table>
The number of thread groups to be dispatched in the Y dimension (max y + 1
<table><tr><th>Value</th><th>Name</th></tr><tr><td>1,FFFFFFFFh</td><td></td></tr></table>
</td></tr>
</table>

GPGPU_DISPATCHDIMZ - GPGPU Dispatch Dimension Z

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02508h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:0</td><td><b>Dispatch Dimension Z</b>
<table><tr><td>Format:</td><td>U32</td></tr></table>
The number of thread groups to be dispatched in the Zdimension (max Z + 1)
<table><tr><th>Value</th><th>Name</th></tr><tr><td>1,FFFFFFFFh</td><td></td></tr></table>
</td></tr>
</table>

GDRST - Graphics Device Reset Control

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	0941Ch

Graphics Device Reset Control Registers

DWord	Bit	Description
0	31:7	**Reserved** Access: RO Reserved
	6:4	**Reserved** Access: RO Reserved.
	3	**Initiate Graphics Blitter Soft Reset** Access: R/W Set Graphics Blitter Soft-Reset Control: '1': Initiate a graphics blitter domain reset. - Cleared by CP once the reset is complete '0': N/A - Once set, clearing of this bit has no effect on CP. Only CP can reset this bit. **Note:** This is a non-posted register.
	2	**Initiate Graphics Media Soft Reset** Access: R/W Set Graphics Media Soft-Reset Control: '1': Initiate a graphics media 0 domain reset. - Cleared by CP once the reset is complete '0': N/A - Once set, clearing of this bit has no effect on CP. Only CP can reset this bit. **Note:** This is a non-posted register.
	1	**Initiate Graphics Render Soft Reset** Access: R/W Set Graphics Render Soft-Reset Control: '1': Initiate a graphics render domain reset. - Cleared by CP once the reset is complete '0': N/A - Once set, clearing of this bit has no effect on CP. Only CP can reset this bit. **Note:** This is a non-posted register.

GDRST - Graphics Device Reset Control

	0	**Initiate Graphics Full Soft Reset** Access: R/W Set Graphics Full Soft-Reset Control: '1': Initiate a full graphics reset (i.e., graphics render, media, and blitter reset). - Cleared by CP once the reset is complete '0': N/A - Once set, clearing of this bit has no effect on CP. Only CP can reset this bit. **Note:** This is a non-posted register.

GFX_ENG_FR - Graphics Engine Fault Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04094h-04097h

Graphics Engine Fault Register

DWord	Bit	Description
0	31:12	**Virtual Address of Fault** Default Value: 00000h Access: R/W This is the original Address of the Page that generated the First fault for this engine. This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW
	11	**GTTSEL** Default Value: 0b Access: R/W This bit indicates if the valid bit happened while using PPGTT or GGTT: 0 - PPGTT, 1 - GGTT This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW
	10:3	**SRCID of Fault** Default Value: 00h Access: R/W This is the Source ID of the unit that requested the cycle that generated the First Page fault for this engine. This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW
	2:1	**Fault Type** Default Value: 00b

GFX_ENG_FR - Graphics Engine Fault Register

<table>
<tr><td></td><td></td><td>
<table><tr><td>Access:</td><td>R/W</td></tr></table>
Type of Fault recorded:
00 - Page Fault.
01 - Invalid PD Fault
10 - Unloaded PD Fault
11 - Invalid and Unloaded PD fault
This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW
</td></tr>
<tr><td></td><td>0</td><td>
<b>Valid Bit</b>
<table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>
This bit indicates that the first fault for this engine has been recorded. It can only be cleared by SW, which will also clear the other fields.
</td></tr>
</table>

FENCE - Graphics Memory Fence Table Register

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	100000h-100007h
Name:	FENCE_0
Address:	100008h-10000Fh
Name:	FENCE_1
Address:	100010h-100017h
Name:	FENCE_2
Address:	100018h-10001Fh
Name:	FENCE_3
Address:	100020h-100027h
Name:	FENCE_4
Address:	100028h-10002Fh
Name:	FENCE_5
Address:	100030h-100037h
Name:	FENCE_6
Address:	100038h-10003Fh
Name:	FENCE_7
Address:	100040h-100047h
Name:	FENCE_8
Address:	100048h-10004Fh
Name:	FENCE_9
Address:	100050h-100057h
Name:	FENCE_10
Address:	100058h-10005Fh
Name:	FENCE_11
Address:	100060h-100067h
Name:	FENCE_12
Address:	100068h-10006Fh
Name:	FENCE_13

FENCE - Graphics Memory Fence Table Register

Address:	100070h-100077h
Name:	FENCE_14
Address:	100078h-10007Fh
Name:	FENCE_15

The graphics device performs address translation from linear space to tiled space for a CPU access to graphics memory (See Memory Interface Functions chapter for information on these memory layouts) using the fence registers. Note that the fence registers are used only for CPU accesses to gfx memory. Graphics rendering/display pipelines use Per Surface Tiling (PST) parameters (found in SURFACE_STATE - see the Sampling Engine chapter) to access tiled gfx memory.

The intent of tiling is to locate graphics data that are close (in X and Y surface axes) in one physical memory page while still locating some amount of line oriented data sequentially in memory for display efficiency. All 3D rendering is done such that the QWords of any one span are all located in the same memory page, improving rendering performance. Applications view surfaces as linear, hence when the cpu access a surface that is tiled, the gfx hardware must perform linear to tiled address conversion and access the correct physical memory location(s) to get the data.

Tiled memory is supported for rendering and display surfaces located in graphics memory. A tiled memory surface is a surface that has a width and height that are subsets of the tiled region's pitch and height. The device maintains the constants required by the memory interface to perform the address translations. Each tiled region can have a different pitch and size. The CPU-memory interface needs the surface pitch and tile height to perform the address translation. It uses the GMAddr (PCI-BAR) offset address to compare with the fence start and end address, to determine if the rendering surface is tiled. The tiled address is generated based on the tile orientation determined from the matching fence register. Fence ranges are at least 4 KB aligned. Note that the fence registers are used only for CPU accesses to graphics memory.

A Tile represents 4 KB of memory. Tile height is 8 rows for X major tiles and 32 rows for Y major tiles. Tile Pitch is 512Bs for X major tiles and 128Bs for Y major tiles. The surface pitch is programmed in 128B units such that the pitch is an integer multiple of "tile pitch".

Engine restrictions on tile surface usage are detailed in Surface Placement Restrictions (Memory Interface Functions). Note that X major tiles can be used for Sampler, Color, Depth, motion compensation references and motion compensation destination, Display, Overlay, GDI Blt source and destination surfaces. Y major tiles can be used for Sampler, depth, color and motion compensation assuming they do not need to be displayed. GDI Blit operations, overlay and display cannot used Tiled Y orientations.

A "PST" graphics surface that will also be accessed via fence needs its base address to be tile row aligned.

Hardware handles the flushing of any pending cycles when software changes the fence upper/lower bounds.

Fence Table Registers occupy the address range specified above. Each Fence Table Register has the following format.

FENCE registers are not reset by a graphics reset. They will maintain their values unless a full chipset reset is performed.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>63:44</td><td><b>Fence Upper Bound</b>
<table><tr><td>Format:</td><td>GraphicsAddress[31:12]</td></tr></table>
Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region).
Graphics Address is the offset within GMADR space.</td></tr>
<tr><td></td><td>43:42</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>41:32</td><td><b>Fence Pitch</b>
<table><tr><td>Format:</td><td>U10-1 Width in 128 bytes</td></tr></table>
This field specifies the width (pitch) of the fence region in multiple of "tile width". For Tile X this field must be programmed to a multiple of 512B ("003" is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B ("000" is the minimum value).

000h = 128B
001h = 256B
...
3FFh = 128KB</td></tr>
<tr><td></td><td>31:12</td><td><b>Fence Lower Bound</b>
<table><tr><td>Format:</td><td>GraphicsAddress[31:12]</td></tr></table>
Bits 31:12 of the starting Graphics Address of the fence region. Fence regions must be aligned to 4KB. This address represents the first 4KB page of the fence region (Lowe Bound is included in the fence region).
Graphics Address is the offset within GMADR space.</td></tr>
<tr><td></td><td>11:2</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>1</td><td><b>Tile Walk</b>
This field specifies the spatial ordering of QWords within tiles.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td>MI_TILE_XMAJOR</td><td>Consecutive SWords (32 Bytes) sequenced in the X direction</td></tr>
<tr><td>1h</td><td>MI_TILE_YMAJOR</td><td>Consecutive OWords (16 Bytes) sequenced in the Y direction</td></tr>
</table></td></tr>
<tr><td></td><td>0</td><td><b>Fence Valid</b>
<table><tr><td>Format:</td><td>MI_FenceValid</td></tr></table>
This field specifies whether or not this fence register defines a fence region.
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0h</td><td>MI_FENCE_INVALID</td></tr>
<tr><td>1h</td><td>MI_FENCE_VALID</td></tr>
</table></td></tr>
</table>

GFX_MODE - Graphics Mode Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000800
Size (in bits):	32
Trusted Type:	1
Address:	0229Ch

Description
This register contains a control bit for the new 2-level PPGTT functions.
DefaultValue = 00002800h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Mask Bits</b><table><tr><td>Format:</td><td>Mask[15:0]</td></tr></table>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td></tr>
<tr><td></td><td>15</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>14</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>13</td><td><b>Flush TLB invalidation Mode</b><table><tr><td>Format:</td><td>U1</td></tr></table>This field controls the invalidation if the TLB cache inside the hardware. When enabled this bit limits the invalidation of the TLB only to batch buffer boundaries, to pipe_control commands which have the TLB invalidation bit set and sync flushes. If disabled, the TLB caches are flushed for every full flush of the pipeline.</td></tr>
<tr><td></td><td>12</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>11</td><td><b>Replay Mode</b><table><tr><td>Format:</td><td>U1 Context Switch Granularity</td></tr></table>This field controls the granularity of the replay mechanism when coming back into a previously preempted context.<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td>mid-triangle preemption</td><td>Super span Level. Pipeline is not flushed. This implies commands parsed are executed speculatively and may not complete before a context switch.</td></tr><tr><td>1h</td><td>mid-cmdbuffer preemption <b>[Default]</b></td><td>Drawcall Level. Pipeline is flushed before switching to the next context. Commands parsed are committed to completing before a context switch.</td></tr></table><table><tr><th>Programming Notes</th></tr></table></td></tr>
</table>

GFX_MODE - Graphics Mode Register

<table>
<tr><td></td><td></td><td><table><tr><td>A fixed function pipe flush is required before modifying this field. Unless pre-emption at a mid-triangle is required the bit must be set.</td></tr></table></td></tr>
<tr><td></td><td>10</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>9</td><td><b>Per-Process GTT Enable</b><table><tr><td>Format:</td><td>Enabled</td></tr></table>Per-Process GTT Enable<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td>PPGTT Disable <b>[Default]</b></td><td>When clear, the Global GTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td></tr><tr><td>1h</td><td>PPGTT Enable</td><td>When set, the PPGTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space. The PD Offset and PD Cacheline Valid registers must be set in all pipes (blitter, MFX, render) before any workload is submitted to hardware. This mode enables support for big pages (32k).</td></tr></table></td></tr>
<tr><td></td><td>8</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>7</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>6:1</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>0</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
</table>

GS_INVOCATION_COUNT - GS Invocation Counter

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02328h

This register stores the number of objects that are part of geometry shader threads. This register is part of the context save and restore.

DWord	Bit	Description
0	63:0	**GS Invocation Count** Number of objects that are dispatched as a geometry shader threads invoked by the GS stage. Updated only when **Statistics Enable** is set in GS_STATE (see the Geometry Shader Chapter in the 3D Volume.)

GS_PRIMITIVES_COUNT - GS Primitives Counter

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02330h

This register reflects the total number of primitives that have been output by the Geometry Shader stage. This register is part of the context save and restore.

DWord	Bit	Description
0	63:0	**GS Primitives Count** Total number of primitives output by the geometry stage. Updated only when Statistics Enable is set in GS_STATE (see the Geometry Shader Chapter in the 3D Volume.)

FLRCTLMSG - GT Function Level Reset Control Message

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	08100h

GT FLR Control Register

DWord	Bit	Description
0	31:16	**Message Mask** Access: RO Message Mask In order to write to bits 15:0, the corresponding message mask bits must be written. For example, for bit 14 to be set, bit 30 needs to be 1: 40004000
	15:1	**Reserved** Access: RO Reserved
	0	**Initiate GT Function Level Reset Message** Access: R/W Set GT Function Level Reset (FLR) 1: Initiate GT FLR - This is a Non-Posted message to reset Render, Media, Blitter, and GTI-Device domains. - This bit is cleared by the CPunit upon completion of the reset.

GT_MODE - GT Mode Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	07008h

Description
This Register is used to control the 6EU and 12EU configuration for GT. Writing 0x01FF01FF to this register enables the 6EU mode.
RegisterType = MMIO_SVL

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Mask Bits</b><table><tr><td>Format:</td><td>Mask[15:0]</td></tr></table>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td></tr>
<tr><td></td><td>15</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>14:13</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>12:11</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>10</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>9</td><td><b>WIZ Hashing Mode High Bit</b><table><tr><td>Format:</td><td>U1</td></tr></table>This field adds additional hashing modes in combination with the WIZ Hashing Mode field. The Value column in the table below refers to this field (high bit) and the WIZ Hashing Mode field (low bit).
This field is don't care if the Hashing Disable bit is set.<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td><b>[Default]</b></td><td>8x8 Checkerboard hashing</td></tr><tr><td>1h</td><td></td><td>8x4 Checkerboard hashing</td></tr><tr><td>2h</td><td></td><td>16x4 Checkerboard hashing</td></tr><tr><td>3h</td><td></td><td>Reserved</td></tr></table><table><tr><th>Programming Notes</th></tr></table></td></tr>
</table>

GT_MODE - GT Mode Register

		8x4 hashing preferred for when msaa enabled
	8	**Reserved** Format: MBZ
	7	**WIZ Hashing Mode** Format: U1 **Description** This field configures the Hashing mode in Windower. This field is don't care if the Hashing Disable bit is set. The WIZ Hashing Mode High Bit field is combined with this field to enable additional modes.
	6:3	**Reserved** Format: MBZ
	2	**Reserved** Format: MBZ
	1	**Reserved** Format: MBZ
	0	**Reserved**

HWSTAM - Hardware Status Mask Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0xFFFFFFFF
Access:	R/W,RO
Size (in bits):	32
Trusted Type:	1
Address:	02098h

The HWSTAM register has the same format as the Interrupt Control Registers. The bits in this register are mask bits that prevent the corresponding bits in the Interrupt Status Register from generating a Hardware Status Write (PCI write cycle). Any unmasked interrupt bit (HWSTAM bit set to 0) will allow the Interrupt Status Register to be written to the ISR location (within the memory page specified by the Hardware Status Page Address Register) when that Interrupt Status Register bit changes state.

Programming Notes

- To write the interrupt to the HWSP, the corresponding IMR bit must also be clear (enabled).
- At most 1 bit can be unmasked at any given time.

DWord	Bit	Description
0	31:0	**Hardware Status Mask Register** Default Value: FFFFFFFFh Format: Array of Masks Refer to the Interrupt Control Register section for bit definitions. Reserved bits are RO.

HWS_PGA - Hardware Status Page Address Register

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1

Address:	22080h
Name:	BCS Hardware Status Page Address Register
ShortName:	BCS_HWS_PGA

Address:	04080h
Name:	RCS Hardware Status Page Address Register
ShortName:	RCS_HWS_PGA

Address:	12080h
Name:	VCS Hardware Status Page Address Register
ShortName:	VCS_HWS_PGA

This register is used to program the 4 KB-aligned System Memory address of the Hardware Status Page used to report hardware status into (typically cacheable) System Memory.

Programming Notes
If this register is written, a workload must subsequently be dispatched to the Render command streamer.

DWord	Bit	Description
0	31:12	**Address** Format: GraphicsAddress[31:12] This field is used by SW to specify Bits 31:12 of the 4 KB-aligned System Memory address of the 4 KB page known as the Hardware Status Page. The Global GTT is used to map this page from the graphics virtual address to physical address.
	11:0	**Reserved** Format: MBZ

HS_INVOCATION_COUNT - HS Invocation Counter

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02300h

This register stores the number of patch objects processed by the HS unit. E.g., A PATCHLIST_2 topology with 6 vertices would cause this counter to increment by 3 (there are 3 2-vertex patch objects in that topology). This register is part of the context save and restore.

DWord	Bit	Description
0	63:0	**HS Invocation Count** Number of patch objects processed by the HS stage. Updated only when HS Enable and HS Statistics Enable are set in 3DSTATE_HS

IA_VERTICES_COUNT - IA Vertices Count

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02310h

This register stores the count of vertices processed by VF. This register is part of the context save and restore.

DWord	Bit	Description
0	63:0	**IA Vertices Count Report** Total number of vertices fetched by the VF stage. This count is updated for every input vertex as long as Statistics Enable is set in VF_STATE (see the Vertex Fetch Chapter in the 3D Volume.)

INSTPM - Instruction Parser Mode Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W,RO
Size (in bits):	32
Trusted Type:	1
Address:	020C0h

The INSTPM register is used to control the operation of the Instruction Parser. Certain classes of instructions can be disabled (ignored) - often useful for detecting performance bottlenecks. Also, Synchronizing Flush operations can be initiated - useful for ensuring the completion (vs. only parsing) of rendering instructions.

Programming Notes

- If an instruction type is disabled, the parser will read those instructions but not process them.
- Error checking will be performed even if the instruction is ignored.
- All Reserved bits are implemented.
- This Register is saved and restored as part of Context.

DWord	Bit	Description
0	31:16	**Mask Bits** Format: Mask[15:0] Masks: These bits serve as write enables for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.
	15	**Reserved**
	14:13	**Reserved** Format: MBZ
	12	**Reserved**
	11	**CLFLUSH Toggle** Access: RO Format: U1 This bit changes polarity each time the MI_CLFLUSH command completes. This bit is Read Only.
	10	**Reserved** Format: MBZ
	9	**TLB Invalidate** Format: U1 If set, this bit allows the command stream engine to invalidate the 3D render TLBs. This bit is valid only with the Sync flush enable. Note: GFX soft resets do not invalidate TLBs, it is up to GFX

INSTPM - Instruction Parser Mode Register

driver to explicitly invalidate TLBs post reset.

8 **Memory Sync Enable**

Format:	U1

If set, this bit allows the command stream engine to write out the data from the local caches to memory. This bit is valid only with the Sync flush enable

7 **Force Sync Command Ordering**

Format:	Enable

Description
By default, driver/OS synchronization commands (MI_STORE_DATA_IMM, for instance) can execute out of order with respect to 3D state and 3D primitive commands. When set, this bit forces ordering of these commands. See section 3.2.2 for a list of these commands.
This bit should be programmed to 1.
This bit should be programmed to 1.
This bit should be programmed to 1.

Value	Name
0b	**[Default]**
1b	

6 **CONSTANT_BUFFER Address Offset Disable**

Format:	Disable

When this bit is clear, the 3DSTATE_CONSTANT_* Buffers' Starting Address is used as a DynamicStateOffset. That is, it serves as an offset from the Dynamic State Base Address. Accesses will be subject to Dynamic State bounds checking.
When this bit is set, the 3DSTATE_CONSTANT_* Buffers' Starting Address is used as a true GraphicsAddress (not an offset). No bounds checking will be performed during access.

5 **Sync Flush Enable**

Format:	U1

This field is used to request a Sync Flush operation. The device will automatically clear this bit before completing the operation. See Sync Flush (Programming Environment).

Programming Notes
• The command parser must be stopped prior to issuing this command by setting the Stop Rings bit in register MI_MODE. Only after observing Rings Idle set in MI_MODE can a Sync Flush be issued by setting this bit. Once this bit becomes clear again, indicating flush complete, the command parser is re-enabled by clearing Stop Rings.

4 **Reserved**

INSTPM - Instruction Parser Mode Register

	3	**Media Instruction Disable** Format: | U1 This bit instructs the Renderer instruction parser to parse and error-check Media instructions, but not execute them. Format = Disable
	2	**3D Rendering Instruction Disable** Format: | U1 This bit instructs the Renderer instruction parser to parse and error-check 3D Rendering instructions, but not execute them. This bit must always be set by software if 3D State Instruction Disable is set. Setting this bit without setting 3D State Instruction Disable is allowed. Format = Disable
	1	**Reserved**
	0	**Texture Palette Load Instruction Disable** Format: | U1 This bit instructs the Renderer instruction parser to parse and error-check Texture Palette Load instructions, but not execute them. Format = Disable

IMR - Interrupt Mask Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0xFFFFFFFF
Access:	R/W,RO
Size (in bits):	32
Address:	020A8h

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

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:0</td><td><b>Interrupt Mask Bits</b>
<table><tr><td>Format:</td><td>InterruptMask[32] Refer to the Interrupt Control Register section for bit definitions.</td></tr></table>
This field contains a bit mask which selects which interrupt bits (from the ISR) are reported in the IIR. Reserved bits in the Interrupt Control Register are RO.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>FFFF FFFFh</td><td><b>[Default]</b></td><td></td></tr>
<tr><td>0h</td><td>Not Masked</td><td>Will be reported in the IIR</td></tr>
<tr><td>1h</td><td>Masked</td><td>Will not be reported in the IIR</td></tr>
</table>
</td></tr>
</table>

L3_LRA_0 - L3 LRA 0

Register Space:	MMIO: 0/2/0
Default Value:	0x3F201F00
Size (in bits):	32
Address:	0403Ch-0403Fh

L3 LRA 0

DWord	Bit	Description
0	31:24	**L3 LRA1 Max** Default Value: 00111111b Access: R/W L3 LRA1 Max Format: U6 Maximum value of programmable LRA1
	23:16	**L3 LRA1 Min** Default Value: 00100000b Access: R/W L3 LRA1 Min Format: U6 Minimum value of programmable LRA1
	15:8	**L3 LRA0 Max** Default Value: 00011111b Access: R/W L3 LRA0 Max Format: U6 Maximum value of programmable LRA0
	7:0	**L3 LRA0 Min** Default Value: 00000000b Access: R/W L3 LRA0 Min Format: U6 Minimum value of programmable LRA1

L3_LRA_1 - L3 LRA 1

Register Space:	MMIO: 0/2/0
Default Value:	0x0900FF40
Size (in bits):	32
Address:	04040h-04043h

L3 LRA 1

DWord	Bit	Description
0	31:30	**Reserved** Default Value: 00b Access: RO Reserved Bits
	29:28	**DC** Default Value: 00b Access: R/W Which LRA should DC use
	27:26	**TEXTURE** Default Value: 10b Access: R/W Which LRA should TEXTURE use
	25:24	**L3** Default Value: 01b Access: R/W Which LRA should L3 use
	23:16	**Reserved** Default Value: 00000000b Access: RO Reserved Bits

L3_LRA_1 - L3 LRA 1

	15:8	**L3 LRA2 Max** Default Value: 11111111b Access: R/W L3 LRA2 Max Format: U6 Maximum value of programmable LRA2
	7:0	**L3 LRA2 Min** Default Value: 01000000b Access: R/W L3 LRA2 Min Format: U6 Minimum value of programmable LRA2

3DPRIM_BASE_VERTEX - Load Indirect Base Vertex

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02440h-02443h

DWord	Bit	Description
0	31:0	**Base Vertex** Format: S31 This register is used to store the Base Vertex of the 3D_PRIMITIVE command when Load Indirect Enable is set.

3DPRIM_INSTANCE_COUNT - Load Indirect Instance Count

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02438h-0243Bh

DWord	Bit	Description
0	31:0	**Instance Count** This register is used to store the Instance Count of the 3D_PRIMITIVE command when Load Indirect Enable is set.

3DPRIM_START_INSTANCE - Load Indirect Start Instance

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	0243Ch-0243Fh

DWord	Bit	Description
0	31:0	**Start Vertex** Format: U32 This register is used to store the Start Instance of the 3D_PRIMITIVE command when Load Indirect Enable is set.

3DPRIM_START_VERTEX - Load Indirect Start Vertex

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02430h-02433h

DWord	Bit	Description
0	31:0	**Start Vertex** Format: U32 This register is used to store the Start Vertex of the 3D_PRIMITIVE command when Load Indirect Enable is set.

3DPRIM_VERTEX_COUNT - Load Indirect Vertex Count

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02434h-02437h

DWord	Bit	Description
0	31:0	**Vertex Count** Format: U32 This register is used to store the Vertex Count of the 3D_PRIMITIVE command when Load Indirect Enable is set.

ERROR - Main Graphic Arbiter Error Report

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	040A0h-040A3h

This register is used to report different error conditions. Error bits are writable.

DWord	Bit	Description
0	31:16	**Reserved** Default Value: 0000h Access: RO Reserved Bits
	15	**Reserved Error Bits 15** Default Value: 0b Access: R/W Reserved Error bits (Future expansion)
	14	**Reserved Error Bits 14** Default Value: 0b Access: R/W Reserved Error bits (Future expansion)
	13	**Reserved Error Bits 13** Default Value: 0b Access: R/W Reserved Error bits (Future expansion)
	12	**Reserved Error Bits 12** Default Value: 0b Access: R/W Reserved Error bits (Future expansion)

ERROR - Main Graphic Arbiter Error Report

	11	**Reserved Error Bits 11** Default Value: 0b Access: R/W Reserved Error bits (Future expansion)
	10	**Reserved Error Bits 10** Default Value: 0b Access: R/W Reserved Error bits (Future expansion)
	9	**Reserved Error Bits 9** Default Value: 0b Access: R/W Reserved Error bits (Future expansion)
	8	**Unloaded PD Error** Default Value: 0b Access: R/W Unloaded PD error The Cache Line containing a PD entry being accessed, was marked as invalid in the last PD load cycle.
	7	**Reserved Error Bits 7** Default Value: 0b Access: R/W Reserved Error bits (Future expansion)
	6	**Reserved**
	5	**Reserved**
	4	**Reserved**

ERROR - Main Graphic Arbiter Error Report

	3	**Reserved**
	2	**Invalid Page Directory Entry Error** Default Value: 0b Access: R/W Invalid Page Directory entry error PD entry’s valid bit is 0
	1	**Reserved**
	0	**TLB Page Fault Error** Default Value: 0b Access: R/W TLB Page Fault error A TLB Page’s GTT translation generated a page fault (GTT entry not valid)

GFX_ARB_ERROR_RPT - Main Graphic Arbiter Error Report Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	040A0h

This register is used to report error conditions. Error bits are writable.

DWord	Bit	Description
0	31:16	**Reserved** Format: MBZ
	15:9	**Reserved**
	8	**Unloaded PD Error** The Cache Line containing a PD entry being accessed was marked as invalid in the last PD load cycle.
	7	**Reserved**
	6	**Reserved**
	5	**Reserved**
	4	**Reserved**
	3	**Hardware Status Page Fault Error** HWSP's GTT translation generated a page fault (GTT entry not valid).
	2	**Invalid Page Directory entry error** PD entry's valid bit is 0.
	1	**Context Page Fault Error** A Context Page's GTT translation generated a page fault (GTT entry not valid).
	0	**TLB Page Fault Error** A TLB Page's GTT translation generated a page fault (GTT entry not valid).

GFX_PEND_TLB_0 - Max Outstanding Pending TLB Requests 0

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04034h-04037h

GFX_PEND_TLB_0 - Max Outstanding Pending TLB Requests 0

DWord	Bit	Description
0	31	**TEX Limit Enable Bit** Default Value: 0b Access: R/W TEX Limit Enable bit Format: U1 This bit is used to enable the pending TLB requests limitation function for the Texture Cache When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	30	**Reserved** Default Value: 0b Access: RO Reserved Format: MBZ
	29:24	**TEX TLB Limit Count** Default Value: 000000b Access: R/W TEX TLB Limit Count Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read
	23	**DC Limit Enable bit** Default Value: 0b Access: R/W

GFX_PEND_TLB_0 - Max Outstanding Pending TLB Requests 0

		DC Limit Enable bit Format: U1 This bit is used to enable the pending TLB requests limitation function for the Instruction Cache. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	22	**Reserved** Default Value: 0b Access: RO Reserved Format: MBZ
	21:16	**DC TLB Limit Count** Default Value: 000000b Access: R/W DC TLB Limit Count Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	15	**VF Limit Enable bit** Default Value: 0b Access: R/W VF Limit Enable bit Format: U1 This bit is used to enable the pending TLB requests limitation function for the Vertex Fetch When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	14	**Reserved** Default Value: 0b Access: RO

GFX_PEND_TLB_0 - Max Outstanding Pending TLB Requests 0

		Reserved Format: MBZ
	13:8	**VF TLB Limit Count** Default Value: 000000b Access: R/W VF TLB Limit Count Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	7	**VMC Limit Enable bit** Default Value: 0b Access: R/W VMC Limit Enable bit Format: U1 This bit is used to enable the pending TLB requests limitation function for the Video Motion Compensation . When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	6	**Reserved** Default Value: 0b Access: RO Reserved Format: MBZ
	5:0	**VMC TLB Limit Count** Default Value: 000000b Access: R/W VMC TLB Limit Count Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.

GFX_PEND_TLB_0 - Max Outstanding Pending TLB Requests 0

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04034h-04037h

DWord	Bit	Description
0	31	**TEX Limit Enable bit** Format: U1 This bit is used to enable the pending TLB requests limitation function for the Texture Cache. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	30	**Reserved** Format: MBZ
	29:24	**TEX TLB Limit Count** Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	23	**ISC Limit Enable bit** Format: U1 This bit is used to enable the pending TLB requests limitation function for the Instruction Cache. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	22	**Reserved** Format: MBZ
	21:16	**ISC TLB Limit Count** Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	15	**VF Limit Enable bit** Format: U1 This bit is used to enable the pending TLB requests limitation function for the Vertex Fetch. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.

GFX_PEND_TLB_0 - Max Outstanding Pending TLB Requests 0

	14	**Reserved** Format: MBZ
	13:8	**VF TLB Limit Count** Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	7	**CS Limit Enable bit** Format: U1 This bit is used to enable the pending TLB requests limitation function for the Command Streamer. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	6	**Reserved** Format: MBZ
	5:0	**CS TLB Limit Count** Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.

GFX_PEND_TLB_1 - Max Outstanding Pending TLB Requests 1

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04038h-0403Bh

GFX_PEND_TLB_1 - Max Outstanding pending TLB requests 1

DWord	Bit	Description
0	31	**SOL Limit Enable bit** Default Value: 0b Access: R/W SOL Limit Enable bit Format: U1 This bit is used to enable the pending TLB requests limitation function for the SOL. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	30	**Reserved** Default Value: 0b Access: RO Reserved Format: MBZ
	29:24	**SOL TLB Limit Count** Default Value: 000000b Access: R/W SOL TLB Limit Count Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	23	**L3 Limit Enable bit** Default Value: 0b Access: R/W L3 Limit Enable bit

GFX_PEND_TLB_1 - Max Outstanding Pending TLB Requests 1

		Format: U1 This bit is used to enable the pending TLB requests limitation function for the L3. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	22	**Reserved** Default Value: 0b Access: RO Reserved Format: MBZ
	21:16	**L3 TLB Limit Count** Default Value: 000000b Access: R/W L3 TLB Limit Count Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	15	**RCZ Limit Enable bit** Default Value: 0b Access: R/W RCZ Limit Enable bit Format: U1 This bit is used to enable the pending TLB requests limitation function for the Render Depth Cache When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	14	**Reserved** Default Value: 0b Access: RO Reserved Format: MBZ

GFX_PEND_TLB_1 - Max Outstanding Pending TLB Requests 1

<table>
<tr><td></td><td>13:8</td><td><b>RCZ TLB Limit Count</b><table><tr><td>Default Value:</td><td>000000b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>RCZ TLB Limit Count

Format: U6

This is the MAX number of Allowed internal pending read requests which require a TLB read.</td></tr>
<tr><td></td><td>7</td><td><b>RCC Limit Enable bit</b><table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>RCC Limit Enable bit

Format: U1

This bit is used to enable the pending TLB requests limitation function for the Render Color Cache. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.</td></tr>
<tr><td></td><td>6</td><td><b>Reserved</b><table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>RO</td></tr></table>Reserved

Format: MBZ</td></tr>
<tr><td></td><td>5:0</td><td><b>RCC TLB Limit Count</b><table><tr><td>Default Value:</td><td>000000b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>RCC TLB Limit Count

Format: U6

This is the MAX number of Allowed internal pending read requests which require a TLB read.</td></tr>
</table>

GFX_PEND_TLB_1 - Max Outstanding Pending TLB Requests 1

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04038h-0403Bh

DWord	Bit	Description
0	31:16	**Reserved** Format: MBZ
	15	**RCZ Limit Enable bit** Format: U1 This bit is used to enable the pending TLB requests limitation function for the Render Depth Cache. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	14	**Reserved** Format: MBZ
	13:8	**RCZ TLB Limit Count** Format: U6 This bit is used to enable the pending TLB requests limitation function for the Render Color Cache. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	7	**RCC Limit Enable bit** Format: U1 This bit is used to enable the pending TLB requests limitation function for the Render Color Cache. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	6	**Reserved** Format: MBZ
	5:0	**RCC TLB Limit Count** Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.

MEDIA_MAX_REQ_COUNT - MAX Requests Allowed - CASC

Register Space:	MMIO: 0/2/0
Default Value:	0x10202020
Size (in bits):	32
Address:	04070h-04073h

Programmable Request Count - CASC

DWord	Bit	Description
0	31:24	**GFX Max Request Limit Count** Default Value: 00010000b Access: R/W This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine . Requests are counted, regardless of kind of cycle (Miss/Hit/Present) Minimum count value must be = 1
	23:16	**MFX/BLT Max Request Limit Count** Default Value: 00100000b Access: R/W This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine . Requests are counted, regardless of kind of cycle (Miss/Hit/Present) Minimum count value must be = 1
	15:14	**Reserved** Default Value: 00b Access: RO Reserved Bits
	13:8	**VLF Max Request Limit Count** Default Value: 100000b Access: R/W This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present) Minimum count value must be = 1
	7:6	**Reserved** Default Value: 00b Access: RO Reserved

GFX_PEND_TLB_1 - Max Outstanding Pending TLB Requests 1

<table>
<tr><td></td><td></td><td>Format: MBZ</td></tr>
<tr><td></td><td>5:0</td><td><b>CASC Max Request Limit Count</b><table><tr><td>Default Value:</td><td>100000b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present)

Minimum count value must be = 1</td></tr>
</table>

GFX_MAX_REQ_COUNT - MAX Requests Allowed - GAM

Register Space:	MMIO: 0/2/0
Default Value:	0x43F20101
Size (in bits):	32
Address:	04074h-04077h

Programmable Request Count - GAM

DWord	Bit	Description
0	31:26	**GAP Writes Max Request Limit Count** Default Value: 010000b Access: R/W This is the MAX number of Allowed Write Requests Count - These counters keep track of the accepted write requests from all GAP clients (RCZ, HiZ,Stc, RCC, L3). Minimum count value must be = 1
	25:20	**CVS Max Request Limit Count** Default Value: 111111b Access: R/W This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present) Minimum count value must be = 1
	19	**Reserved** Default Value: 0b Access: RO Reserved Format: MBZ
	18:13	**L3 Max Request Limit Count** Default Value: 010000b Access: R/W This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present) Minimum count value must be = 1
	12	**Reserved**

GFX_MAX_REQ_COUNT - MAX Requests Allowed - GAM

		Default Value: 0b Access: RO Reserved Format: MBZ
	11:6	**Z Request Limit Count** Default Value: 000100b Access: R/W This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present) Minimum count value must be = 1
	5:0	**RCC Request Limit Count** Default Value: 000001b Access: R/W This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present) Minimum count value must be = 1

GAC_ERROR - Media Arbiter Error Report Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	140A0h

These registers are directly mapped for the Error Reporting bits.

DWord	Bit	Description
0	31:11	**Reserved/ECO**
	10	**Reserved**
	9	**Reserved**
	8	**Unloaded PD error** The Cache Line containing a PD entry being accessed was marked as invalid in the last PD load cycle.
	7	**Reserved**
	6	**Reserved**
	5	**Reserved**
	4	**Reserved**
	3	**Reserved**
	2	**Reserved**
	1	**Context Page Fault Error** A Context Page's GTT translation generated a page fault (GTT entry not valid)
	0	**TLB Page Fault Error** A TLB Page's GTT translation generated a page fault (GTT entry not valid)

MEDIA_ENG_FR - Media Engine Fault Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04194h-04197h

Media Engine Fault Register

DWord	Bit	Description
0	31:12	**Virtual Address of Fault** Default Value: 00000h Access: R/W This is the original Address of the Page that generated the First fault for this engine. This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW
	11	**GTTSEL** Default Value: 0b Access: R/W This bit indicates if the valid bit happened while using PPGTT or GGTT: 0 - PPGTT, 1 - GGTT This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW
	10:3	**SRCID of Fault** Default Value: 00h Access: R/W This is the Source ID of the unit that requested the cycle that generated the First Page fault for this engine. This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW
	2:1	**Fault Type** Default Value: 00b Access: R/W Type of Fault recorded: 00 - Page Fault. 01 - Invalid PD Fault

MEDIA_ENG_FR - Media Engine Fault Register

<table>
<tr><td></td><td></td><td>10 - Unloaded PD Fault
11 - Invalid and Unloaded PD fault
This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW</td></tr>
<tr><td></td><td>0</td><td><b>Valid Bit</b><table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>This bit indicates that the first fault for this engine has been recorded. It can only be cleared by SW, which will also clear the other fields.</td></tr>
</table>

AVC_CABAC_INSERTION_COUNT - MFC_AVC_CABAC_INSERTION_COUNT

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124ACh

This register stores the count in bytes of **CABAC ZERO_WORD insertion. It is primarily provided for statistical data gathering.**

DWord	Bit	Description
0	31:0	**MFC AVC Cabac Insertion Count** Total number of bytes in the bitstream output before for the CABAC zero word insertion. This count is updated each time when the insertion count is incremented.

MFC_VIN_AVD_ERROR_CNTR - MFC_AVC Bitstream Decoding Front-End Parsing Logic Error Counter

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	12804h

DWord	Bit	Description
0 avd_error_flagsR[31:0]	31:0	**Reserved** Format: MBZ

MFC_IMAGE_STATUS_CONTROL - MFC Image Status Control

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124B8h

This register stores the suggested data for next frame in multi-pass.

DWord	Bit	Description
0	31:24	**Cumulative slice delta QP**
	23:16	**QP Value** suggested slice QP delta value for frame level Rate control. This value can be +ve or -ve
	15	**QP-Polarity Change** Cumulative slice delta QP polarity change.
	14:13	**Num-Pass Polarity Change** Number of passes after cumulative slice delta QP polarity changes.
	12	**Reserved** Format: MBZ
	11:8	**Total Num-Pass**
	7:4	**Reserved** Format: MBZ
	3	**Reserved** Format: MBZ
	2	**Panic** Panic triggered to avoid too big packed file.
	1	**Frame Bit Count** Frame Bit count over-run/under-run flag
	0	**Max Conformance Flag** Max Macroblock conformance flag or Frame Bit count over-run/under-run

MFC_IMAGE_STATUS_MASK - MFC Image Status Mask

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124B4h

This register stores the image status(flags).

DWord	Bit	Description
0	31:0	**Control Mask** Control Mask for dynamic frame repeat.

MFC_QUP_CT - MFC QP Status Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124BCh

This register stores the suggested QP COUNTS in multi-pass.

DWord	Bit	Description
0	31:24	**Cumulative QP Adjust** Format: U8 Cumulative QP adjustment after multiple passes. If there is no need to multi-pass, this value would be zero. (This is in sign magnitude form).
	23:0	**Cumulative QP** Format: U24 Cumulative QP for all MB of a Frame (Can be used for computing average QP).

MFD_ERROR_STATUS - MFD Error Status

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12400h

This register stores the error status flags and count reports by the bit-stream decoder.
This register is not part of hardware context save and restore. Driver should read the content prior to starting a new batch/frame.

DWord	Bit	Description
0	31:16	**Number of Error Events** Exists If: //JPEG == True Format: U16 This 16-bit field indicates the number of error events detected during decoding the current frame. This field is clear at the start of decoding a new frame.
	31:16	**Number of MB Concealment** Exists If: //AVC CAVLC, AVC CABAC, VC1 and MPEG2 == True Format: U16 This 16-bit field indicates the number of MB is concealmed by hardware. This field is clear at the start of decoding a new frame.
	15:0	**Bit-stream Error flags** Bit-stream error detected by the VLD bit-steram decoder. These flags are reset at the beginning of a frame and updated until starting of another frame. AVC CAVLC: Please refer to AVC CAVLC table for each bit field AVC CABAC: Please refer to AVC CABAC table for each bit field VC1: Please refer to VC1 table for each bit field MPEG2: Please refer to MPEG2 table for each bit field JPEG: Please refer to JPEG table for each bit field

MFD_PICTURE_PARAM - MFD Picture Parameter

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	12420h

DWord	Bit	Description
0	31:0	**Reserved** Format: MBZ

MFX_LAT_CT1 - MFX_Memory_Latency_Count1

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12470h

This register stores the max and min memory latency counts reported on reference read requests. This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:24	**Max Request Count** This field indicates the maximum number of requests allowed by the memory sub-system channel.
	23:16	**Current Request Count** This field indicates the number of requests currently outstanding in the memory sub-system. This field should report with a value of zero at the end of frame; otherwise the motion compensation engine is most likely hung waiting for read data to be returned from sub-system.
	15:8	**MFX Reference picture read request - Max Latency Count in 8xMedia clock cycles** This field reports the maximum memory latency count on all reference reads requested by the motion compensation engine.
	7:0	**MFX Reference picture read request - Min Latency Count in 8xMedia clock cycles** This field reports the minimum memory latency count on all reference reads requested by the motion compensation engine.

MFX_SE-BIN_CT - MFX Frame BitStream SE/BIN Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	1246Ch

This register stores the number of BINs (AVC CABAC) and SEs (CAVLD, VLD) decoded in a frame.
This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:0	**MFX Frame Bit-stream SE/BIN Count** Total number of BINs/SEs decoded in current frame. This number is used with frame performance count to derive Bin/clk or SE/clk.

MFX_MB_COUNT - MFX Frame Macroblock Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12468h

This register stores the number of Macro-blocks decoded/encoded in current frame.

This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:0	**MFX Frame Macro-block Count** Total number of Macro-block decoded/encoded in current frame. This number is used with frame performance count to derive clk/mb.

MFX_MISS_CT - MFX Frame Motion Comp Miss Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12488h

This register stores the total number of cacheline hits occurred in the motion compensation cache per frame. This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:16	**Reserved** Format: MBZ
	15:0	**MFX Frame Motion Comp cache miss Count** Total number of CL misses occurred in the 12KB cache of the motion compensation engine per frame. This number is used along with **MFX Frame Motion Comp Read Count** to derive motion comp cache miss/hit ratio.

MFX_READ_CT - MFX Frame Motion Comp Read Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12484h

This register stores the total number of reference picture read requests made by the Motion Compensation engine per frame.

This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:20	**Reserved** Format: MBZ
	19:0	**MFX Frame Motion Comp CL read request Count** Total number of reference picture read requests by the motion compensation engine per frame.

MFX_FRAME_PERFORMANCE_CT - MFX Frame Performance Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12460h

This register stores the number of clock cycles spent decoding/encoding the current frame.
This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:0	**MFX Frame Performance Count** Total number of clocks between frame start and frame end. This count is incremented on crm_clk

MFX_ROW-PER-BS_COUNT - MFX Frame Row-Stored/BitStream Read Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12480h

This register stores the total number of row-stored/bit-stream read requests made by the pre-fetch engine per frame.
This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:16	**Reserved** Format: MBZ
	15:0	**MFX row-stored/bit-stream read request Count** Total number of row-stored/bit-stream read requests sent by the memory pre-fetch engine per frame.

MFX_LAT_CT2 - MFX Memory Latency Count2

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12474h

This register stores the accumulative memory latency count on reference picture read requests.
This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:26	**Reserved** Format: MBZ
	25:0	**MFX Reference picture read request - Accumulative Memory Latency Count for the entire frame in 8xMedia clock cycles** The accumulative memory latency count of all reference reads requested by motion compensative engine per frame. This number is used with **MFX Frame Motion Comp Read Count** to derive average memory latency.

MFX_LAT_CT3 - MFX Memory Latency Count3

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12478h

This register stores the max and min memory latency counts reported on row-stored/bit-stream read requests. Max and current requests into memory sub-system engine.
This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:24	**Max Request Count** This field indicates the maximum number of requests allowed by the memory sub-system channel.
	23:16	**Current Request Count** This field indicates the number of requests currently outstanding in the memory sub-system. This field should report with a value of zero at the end of frame; otherwise the pre-fetch engine most likely hung waiting for read data to be returned from sub-system.
	15:8	**MFX row-stored/bit-stream read request - Max Latency Count in 8xMedia clock cycles** This field reports the maximum memory latency count on all row-stored/bit-stream reads requested by the memory pre-fetch engine.
	7:0	**MFX row-stored/bit-stream read request - Min Latency Count in 8xMedia clock cycles** This field reports the minimum memory latency count on all row-stored/bit-stream reads requested by the memory pre-fetch engine.

MFX_LAT_CT4 - MFX Memory Latency Count4

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	1247Ch

This register stores the accumulative memory latency count on row-stored/bit-stream read requests.
This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:26	**Reserved** Format: MBZ
	25:0	**MFX row-stored/bit-stream read request - Accumulative Memory Latency Count for the entire frame in 8xMedia clock cycles** The accumulative memory latency count of all row-stored/bit-stream reads requested by pre-fetch engine per frame. This number is used with **Frame row-stored/bit-stream memory read count** to derive average memory latency.

MFX_STATUS_FLAGS - MFX Pipeline Status Flags

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12438h

This register stores the various pipeline status flags.
This register is not part of hardware context save and restore.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:17</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>16</td><td><b>MFX Active</b>
Frame decoding/encoding is in progress.
Set on frame_start;
clear on frame_end.</td></tr>
<tr><td></td><td>15:10</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>9</td><td><b>Streamout Enable</b></td></tr>
<tr><td></td><td>8</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>7</td><td><b>Post Deblocking Mode Enable</b></td></tr>
<tr><td></td><td>6</td><td><b>Pre Deblocking Mode Enable</b></td></tr>
<tr><td></td><td>5</td><td><b>Decoder Mode Select</b><table><tr><th>Value</th><th>Name</th></tr><tr><td>0</td><td>Configure the MFD Engine for VLD Mode</td></tr><tr><td>1</td><td>Configure the MFD Engine for IT Mode</td></tr></table></td></tr>
<tr><td></td><td>4</td><td><b>Codec Select</b><table><tr><th>Value</th><th>Name</th></tr><tr><td>0</td><td>Decode</td></tr><tr><td>1</td><td>Encode</td></tr></table></td></tr>
<tr><td></td><td>3:2</td><td><b>Video Mode</b><table><tr><th>Value</th><th>Name</th></tr><tr><td>00b</td><td>MPEG2</td></tr><tr><td>01b</td><td>VC1</td></tr><tr><td>10b</td><td>AVC</td></tr><tr><td>11b</td><td>JPEG</td></tr></table></td></tr>
</table>

MFX_STATUS_FLAGS - MFX Pipeline Status Flags

<table>
<tr><td></td><td>1</td><td><b>Decoder Short Format Mode</b>
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td></td><td>AVC/VC1 Short Format Mode is in use</td></tr>
<tr><td>1</td><td></td><td>AVC/VC1 Long Format Mode is in use</td></tr>
</table></td></tr>
<tr><td></td><td>0</td><td><b>Stitch Mode</b>
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0b</td><td></td><td>Not in Stitch Mode</td></tr>
<tr><td>1b</td><td></td><td>In the Special Stitch Mode</td></tr>
</table></td></tr>
</table>

MFX_SLICE_PERFORM_CT - MFX Slice Performance Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	12464h

This register stores the number of clock cycles spent decoding/encoding the current slice.
This register is not part of hardware context save and restore.

DWord	Bit	Description
0	31:0	**MFX Frame Performance Count** Total number of clocks between slice start and slice end. This count is incremented on crm_clk

MISCCPCTL - Misc. Clocking / Reset Control Registers

Register Space:	MMIO: 0/2/0
Default Value:	0x00000002
Size (in bits):	32
Address:	09424h

Miscellaneous Clocking / Reset Control Registers.

DWord	Bit	Description
0	31:8	**Bonus ECO bits** Access: R/W Bonus ECO bits.
	7:2	**Reserved** Access: RO Reserved.
	1	**L1 Clock Ungate Enabling Control During Reset** Default Value: 1b Access: R/W Control to enable/disable L1 clock gating during soft resets and FLR reset processing: 1 - Disable L1 clock gating during soft resets and FLR. 0 - Enable L1 clock gating during soft resets and FLR (default op).
	0	**DOP Clock Gating Enable for Render Clocks** Access: R/W Controls the Enabling of the DOP-level Render (crclk/cr2xclk) Clock Gating via PM event messages: 1 - Clock gating is enabled. 0 - Clock gating is disabled.

RSTCTL - Misc. Reset Control Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	09420h

Miscellaneous reset control registers.

DWord	Bit	Description
0	31:4	**Reserved** Access: RO Reserved
	3:2	**Reset Staggering Period Control** Access: R/W Reset assertion staggering period between reset domains during FLR and soft-resets: 00: 24 cs clocks 01: 48 cs clocks 10: 72 cs clocks 11: 96 cs clocks
	1:0	**Reset Residency Control** Access: R/W Reset assertion residency period for FLR and soft-resets. 00: 24 cs clocks 01: 48 cs clocks 10: 96 cs clocks 11: 192 cs clocks

GAB_MODE - Mode Register for GAB

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	220A0h-220A3h

The GAB_MODE register
contains information that controls configurations in the GAB.

DWord	Bit	Description
0	31:16	**Reserved** Access: WO This register has bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. To set bit0, for example, the data would be 0x0001_0001. To clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16.
	15:6	**Reserved** Read/Write
	5:3	**BLB Arbitration Priority** Format: U3
	2:0	**BCS Arbitration Priority** Format: U3

GAC_MODE - Mode Register for GAC

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	120A0h-120A3h

The GAC_MODE register contains information that controls configurations in the GAC.

DWord	Bit	Description
0	31:16	**Masks** Format: Mask[15:0] A 1 in a bit in this field allows the modification of the corresponding bit in Bits 15:0.
	15:0	**Reserved** Access: R/W

GAFS_MODE - Mode Register for GAFS

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	0212Ch

DWord	Bit	Description
0	31:16	**Mask Bits** Format: Mask[15:0] Masks: These bits serve as write enables for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.
	15:11	**Reserved** Format: MBZ
	10	**Reserved** Format: MBZ
	9	**Reserved**
	8:2	**Reserved** Format: MBZ
	1	**Reserved**
	0	**Selection of Arbitration for GAFS** Format: MBZ GAFS data return policy from FFROB is a round-robin. This bit freezes the round robin to FF pipeline order.

MTTLB_VA - MT Virtual Page Address Registers

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04800h-04803h

DWord	Bit	Description
0	31:12	**Address** Format: GraphicsAddress[31:12] Page virtual address.
	11:0	**Reserved** Format: MBZ

NOPID - NOP Identification Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Size (in bits):	32
Trusted Type:	1
Address:	02094h

Description
Access: RW
The NOPID register contains the Noop Identification value specified by the last MI_NOOP instruction that enabled this register to be updated.

DWord	Bit	Description
0	31:22	**Reserved** Format: MBZ
	21:0	**Reserved**

PAK_ERR - PAK_Stream-Out Report (Errors)

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124E8h

DWord	Bit	Description
0	31:22	**Reserved** Format: MBZ
	21	**Incorrect IntraMBFlag in I-slice(AVCf)**
	20	**Out of Range Symbol Code(AVC/mpeg2)**
	19	**Incorrect MBType(AVC/mpeg2**
	18	**Motion Vectors are not inside the frame boundary(mpeg2)**
	17	**Scale code is zero(mpeg2)**
	16	**Incorrect DCTtype for given motionType(mpeg2)**
	15:8	**MB Y-position**
	7:0	**MB X-position**

PAK_WARN - PAK_Stream-Out Report (Warnings)

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124E4h

DWord	Bit	Description
0	31:22	**Reserved** Format: MBZ
	21	**Skip Run > 8192 (AVC)**
	20	**Incorrect SkipMB (AVC and mpeg2)**
	19	**Incorrect MV difference for dual-prime MB (mpeg2)**
	18	**End of Slice signal missing on last MB of a Row(mpeg2)**
	17	**Incorrect DCT type for field picture**
	16	**MVs are not within defined range by fcode**
	15:8	**MB Y-position**
	7:0	**MB X-position**

PAK_REPORT_STAT - PAK Report Running Status

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124ECh

DWord	Bit	Description
0	31:1	**Reserved**
	0	**PAK Status**

Value	Name	Description
0		PAK engine is IDLE
1		PAK engine is currently generating bit stream.

PAT_INDEX - PAT Index

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000003
Access:	R/W
Size (in bits):	32

DWord	Bit	Description
0	31:10	**Reserved** Default Value: 0000000000000000000000b
	7:6	**Reserved** Default Value: 00b
	5:4	**LRU AGE** Default Value: 00b 00: Take the age value from Uncore CRs 01: Assign the age of "0" 10: Do not change the age on a hit 11: Assign the age of "3"
	1:0	**Mem Type** Default Value: 11b 00: Uncacheable(UC) 01: Write Combining(WC) 10: Write through(WT) 11: Write back(WB)

PDTLB_VA - PDTLB_VA Virtual page Address Registers

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24B00h-24B03h

This register is directly mapped to the current Virtual Addresses in the PD TLB.

DWord	Bit	Description
0	31:12	**ADDRESS** Format: GraphicsAddress[31:12] PAGE VIRTUAL ADDRESS
	11:0	**RESERVED** Format: MBZ

UHPTR - Pending Head Pointer Register

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02134h
Name:	RCS Pending Head Pointer Register
ShortName:	RCS_UHPTR
Address:	12134h
Name:	VCS Pending Head Pointer Register
ShortName:	VCS_UHPTR
Address:	22134h
Name:	BCS Pending Head Pointer Register
ShortName:	BCS_UHPTR

Programming Notes
RenderCS Only: Once SW uses UHPTR to preempt the existing workload, should explicitly program MI_SET_CONTEXT to save the preempted context status before submitting the new workload.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:3</td><td><b>Head Pointer Address</b>
<table><tr><td>Format:</td><td>GraphicsAddress[31:3]</td></tr></table>
<table><tr><th>Description</th></tr><tr><td>This register represents the GFX address offset where execution should continue in the ring buffer following execution of an MI_ARB_CHECK command.</td></tr></table></td></tr>
<tr><td></td><td>2:1</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>0</td><td><b>Head Pointer Valid</b>
<table><tr><th>Description</th></tr><tr><td>This bit is set by the software to request a pre-emption.</td></tr><tr><td>It is reset by hardware when an MI_ARB_CHECK command is parsed by the command streamer. The hardware uses the head pointer programmed in this register at the time the reset is generated.</td></tr></table>
<table><tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td>InValid</td><td>No valid updated head pointer register, resume execution at the current location in the ring buffer</td></tr>
<tr><td>1</td><td>Valid</td><td>Indicates that there is an updated head pointer programmed in this register</td></tr></table></td></tr>
</table>

PP_DCLV - PPGTT Directory Cacheline Valid Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Size (in bits):	64
Address:	02220h

Description

Access: R/W

This register controls update of the on-chip PPGTT Directory Cache during a context restore. Bits that are set will trigger the load of the corresponding 16 directory entry group. This register is restored with context (prior to restoring the on-chip directory cache itself). This register is also restored when switching to a context whose LRCA matches the current CCID if the **Force PD Restore** bit is set in the context descriptor.
The context image of this register must be updated and maintained by SW; SW should not normally need to read this register.
This register can also effectively be used to limit the size of a process's virtual address space. Any access by a process that requires a PD entry in a set that is not enabled in this register will cause a fatal error, and no fetch of the PD entry will be attempted.

Programming Notes

Page Directory Base Register is a Global Context Register (power context) and not maintained per context in ring buffer mode of submission. One should explicitly load PP_DCLV followed by PP_DIR_BASE register through Load Register Immediate commands in Ring Buffer before submitting a context. One should program these registers after ensuring the pipe is completely flushed with TLB's invalidated.

DWord	Bit	Description
0	63:32	**Reserved** Format: MBZ
	31:0	**PPGTT Directory Cache Restore [1..32] 16 entries** Format: BitMask[Enable] If set, the [1st..32nd] 16 entries of the directory cache are considered valid and will be brought in on context restore. If clear, these entries are considered invalid and fetch of these entries will not be attempted.

PP_PFD[0:31] - PPGTT Page Fault Data Registers

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	04580h

The GTT Page Fault Log entries can be read from these registers.
4580h-4583h: Fault Entry 0
...
45FCh-45FFh: Fault Entry 31

DWord	Bit	Description
0	31:12	**Fault Entry Page Address** Format: GraphicsAddress[31:12] This RO field contains the faulting page address for this Fault Log entry. This field will contain a valid fault address only if the bit in the GTT Page Fault Indication Register corresponding with the address offset of this entry is set.
	11:0	**Reserved** Format: MBZ

MI_PREDICATE_RESULT - Predicate Rendering Data Result

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02418h

DWord	Bit	Description
0	31:1	**Reserved** Format: MBZ
	0	**MI_PREDICATE_RESULT** This bit is the result of the last MI_PREDICATE.

MI_PREDICATE_DATA - Predicate Rendering Data Storage

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Address:	02410h-02417h

DWord	Bit	Description
0	63:0	**MI_PREDICATE_DATA** This register is used either as computed value based off the MI_PREDICATE_SRC0 and MI_PREDICATE_SRC1 or a temporary register. See Predicate Rendering section for more details.

MI_PREDICATE_SRC0 - Predicate Rendering Temporary Register0

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Address:	02400h-02407h

DWord	Bit	Description
0	63:0	**MI_PREDICATE_SRC0** This register is a temporary register for Predicate Rendering. See Predicate Rendering section for more details.

MI_PREDICATE_SRC1 - Predicate Rendering Temporary Register1

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Address:	02408h-0240Fh

DWord	Bit	Description
0	63:0	**MI_PREDICATE_SRC1** This register is a temporary register for Predicate Rendering. See Predicate Rendering section for more details.

IA_PRIMITIVES_COUNT - Primitives Generated By VF

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02318h

This register stores the count of primitives generated by VF. This register is part of the context save and restore.

DWord	Bit	Description
0	63:0	**IA Primitives Count Report** Total number of primitives output by the Vertex Fetch (IA) stage. This count is updated for every primitive output by the VF stage, as long as Statistics Enable is set in VF_STATE (see the Vertex Fetch Chapter in the 3D Volume.)

MIDARB_PRIO_HIT_REGISTER - Priority Field in Programmable Arbitration for Hit

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043A0h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:12</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>11:9</td><td><b>Encoded Programmable Priority for MIDARB_GOTOFIELD_HIT3 Register</b>
<table>
<tr><th>Encoding</th><th>Priority 1</th><th>Priority 2</th><th>Priority 3</th></tr>
<tr><td>000</td><td>CS/VF/ISC</td><td>MT/CTC</td><td>RCC</td></tr>
<tr><td>001</td><td>CS/VF/ISC</td><td>RCC</td><td>MT/CTC</td></tr>
<tr><td>010</td><td>RCC</td><td>CS/VF/ISC</td><td>MT/CTC</td></tr>
<tr><td>011</td><td>RCC</td><td>MT/CTC</td><td>CS/VF/ISC</td></tr>
<tr><td>100</td><td>MT/CTC</td><td>CS/VF/ISC</td><td>RCC</td></tr>
<tr><td>101</td><td>MT/CTC</td><td>RCC</td><td>CS/VF/ISC</td></tr>
<tr><td>110</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>111</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td></td><td>8:6</td><td><b>Encoded Programmable Priority for MIDARB_GOTOFIELD_HIT2 Register</b></td></tr>
<tr><td></td><td>5:3</td><td><b>Encoded Programmable Priority for MIDARB_GOTOFIELD_HIT1 Register</b></td></tr>
<tr><td></td><td>2:0</td><td><b>Encoded Programmable Priority for MIDARB_GOTOFIELD_HIT0 Register</b></td></tr>
</table>

MIDARB_PRIO_NP_REGISTER - Priority Field in Programmable Arbitration for Hit-NP

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	043A4h
Address:	04208h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:20</td><td><b>Reserved</b></td></tr>
<tr><td></td><td>19:15</td><td><b>Encoded Programmable Priority for MIDARB_GOTOFIELD_MISS3 Register</b>
<table>
<tr><th>Encoding</th><th>Priority 1</th><th>Priority 2</th><th>Priority 3</th><th>Priority 4</th></tr>
<tr><td>00000</td><td>CS/VF/ISC</td><td>MT_CTC</td><td>RCC</td><td>RCZ_HiZ_Stnc</td></tr>
<tr><td>00001</td><td>CS/VF/ISC</td><td>RCC</td><td>MT_CTC</td><td>RCZ_HiZ_Stnc</td></tr>
<tr><td>00010</td><td>RCC</td><td>CS/VF/ISC</td><td>MT_CTC</td><td>RCZ_HiZ_Stnc</td></tr>
<tr><td>00011</td><td>RCC</td><td>MT_CTC</td><td>CS/VF/ISC</td><td>RCZ_HiZ_Stnc</td></tr>
<tr><td>00100</td><td>MT_CTC</td><td>CS/VF/ISC</td><td>RCC</td><td>RCZ_HiZ_Stnc</td></tr>
<tr><td>00101</td><td>MT_CTC</td><td>RCC</td><td>CS/VF/ISC</td><td>RCZ_HiZ_Stnc</td></tr>
<tr><td>01000</td><td>CS/VF/ISC</td><td>MT_CTC</td><td>RCZ_HiZ_Stnc</td><td>RCC</td></tr>
<tr><td>01001</td><td>CS/VF/ISC</td><td>RCC</td><td>RCZ_HiZ_Stnc</td><td>MT_CTC</td></tr>
<tr><td>01010</td><td>RCC</td><td>CS/VF/ISC</td><td>RCZ_HiZ_Stnc</td><td>MT_CTC</td></tr>
<tr><td>01011</td><td>RCC</td><td>MT_CTC</td><td>RCZ_HiZ_Stnc</td><td>CS/VF/ISC</td></tr>
<tr><td>01100</td><td>MT_CTC</td><td>CS/VF/ISC</td><td>RCZ_HiZ_Stnc</td><td>RCC</td></tr>
<tr><td>01101</td><td>MT_CTC</td><td>RCC</td><td>RCZ_HiZ_Stnc</td><td>CS/VF/ISC</td></tr>
<tr><td>10000</td><td>CS/VF/ISC</td><td>RCZ_HiZ_Stnc</td><td>MT_CTC</td><td>RCC</td></tr>
<tr><td>10001</td><td>CS/VF/ISC</td><td>RCZ_HiZ_Stnc</td><td>RCC</td><td>MT_CTC</td></tr>
<tr><td>10010</td><td>RCC</td><td>RCZ_HiZ_Stnc</td><td>CS/VF/ISC</td><td>MT_CTC</td></tr>
<tr><td>10011</td><td>RCC</td><td>RCZ_HiZ_Stnc</td><td>MT_CTC</td><td>CS/VF/ISC</td></tr>
<tr><td>10100</td><td>MT_CTC</td><td>RCZ_HiZ_Stnc</td><td>CS/VF/ISC</td><td>RCC</td></tr>
<tr><td>10101</td><td>MT_CTC</td><td>RCZ_HiZ_Stnc</td><td>RCC</td><td>CS/VF/ISC</td></tr>
<tr><td>11000</td><td>RCZ_HiZ_Stnc</td><td>CS/VF/ISC</td><td>MT_CTC</td><td>RCC</td></tr>
<tr><td>11001</td><td>RCZ_HiZ_Stnc</td><td>CS/VF/ISC</td><td>RCC</td><td>MT_CTC</td></tr>
<tr><td>11010</td><td>RCZ_HiZ_Stnc</td><td>RCC</td><td>CS/VF/ISC</td><td>MT_CTC</td></tr>
</table>
</td></tr>
</table>

MIDARB_PRIO_NP_REGISTER - Priority Field in Programmable Arbitration for Hit-NP

<table>
<tr><td></td><td></td><td><table>
<tr><td>11011</td><td>RCZ_HiZ_Stnc</td><td>RCC</td><td>MT_CTC</td><td>CS/VF/ISC</td></tr>
<tr><td>11100</td><td>RCZ_HiZ_Stnc</td><td>MT_CTC</td><td>CS/VF/ISC</td><td>RCC</td></tr>
<tr><td>11101</td><td>RCZ_HiZ_Stnc</td><td>MT_CTC</td><td>RCC</td><td>CS/VF/ISC</td></tr>
<tr><td>Other values</td><td>Reserved</td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td></td><td>14:10</td><td><b>Encoded Programmable Priority for MIDARB_GOTOFIELD_NP2 Register</b></td></tr>
<tr><td></td><td>9:5</td><td><b>Encoded Programmable Priority for MIDARB_GOTOFIELD_NP1 Register</b></td></tr>
<tr><td></td><td>4:0</td><td><b>Encoded Programmable Priority for MIDARB_GOTOFIELD_NP0 Register</b></td></tr>
</table>

MIDARB_PRIO_MISS_REGISTER - Priority Field in Programmable Arbitration for Miss

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04204h

DWord	Bit	Description
0	31:20	**Reserved**
	19:15	**Encoded Programmable Priority for MIDARB_GOTOFIELD_MISS3 Register**
	14:10	**Encoded Programmable Priority for MIDARB_GOTOFIELD_MISS2 Register**
	9:5	**Encoded Programmable Priority for MIDARB_GOTOFIELD_MISS1 Register**
	4:0	**Encoded Programmable Priority for MIDARB_GOTOFIELD_MISS0 Register**

PRIV_PAT - Private PAT

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Size (in bits):	32
Address:	040E8h

DWord	Bit	Description
0	31:0	**Private PAT** Default Value: 00000000h Access: R/W Bit[31:16]: Reserved. Bit[15:8]: PPGTT Private PAT. Bit[7:6]: Reserved. Bit[5:4]: (See below.) 00b: Age is 0. 01b: Age is 1. 10b: Age is 2. 11b: Age is 3. Bit[1:0]: (see below): 00b: Uncached with fence. 01b: Write Combining (traditional UC). 10b: Write Through. 11b: Write Back.

PS_DEPTH_COUNT - PS Depth Count

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02350h

This register stores the value of the count of samples that have passed the depth test. This register is part of the context save and restore. Note that the value of this register can be obtained in a pipeline-synchronous fashion without a pipeline flush by using the 3DCONTROL command. See 3D Overview in the 3D volume.

DWord	Bit	Description
0..1	63:0	**Depth Count** This register reflects the total number of samples that have passed the depth test (i.e., will be visible). All samples are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Samples that pass the depth test but fail the stencil test will not be counted.

PS_INVOCATION_COUNT - PS Invocation Count

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02348h

DWord	Bit	Description
0..1	63:0	**PS Invocation Count** Reflects a count of the total number of pixels (including unlit "helper pixels" within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels.

RCGCTL1 - RAM Clock Gating Control 1

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	09410h

RAM Clock Gating Control Registers.

DWord	Bit	Description
0	31	**USBunit RAM Clock Gating Disable** Access: R/W USBunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	30	**VLFunit RAM Clock Gating Disable** Access: R/W VLFunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	29	**VISunit RAM Clock Gating Disable** Access: R/W VISunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	28	**STCunit RAM Clock Gating Disable** Access: R/W STCunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	27	**TDSunit RAM Clock Gating Disable** Access: R/W TDSunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

RCGCTL1 - RAM Clock Gating Control 1

	26	**VMCunit RAM Clock Gating Disable** Access: | R/W VMCunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	25	**QRCunit RAM Clock Gating Disable** Access: | R/W QRCunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	24	**SCunit RAM Clock Gating Disable** Access: | R/W SCunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	23	**SVLunit RAM Clock Gating Disable** Access: | R/W SVLunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	22	**VFunit RAM Clock Gating Disable** Access: | R/W VFunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	21	**URBunit RAM Clock Gating Disable** Access: | R/W URBunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle

RCGCTL1 - RAM Clock Gating Control 1

		for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	20	**Reserved** Access: | RO Reserved.
	19	**SVGunit RAM Clock Gating Disable** Access: | R/W SVGunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	18	**RCZunit RAM Clock Gating Disable** Access: | R/W RCZunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	17	**RCPBEunit RAM Clock Gating Disable** Access: | R/W RCPBEunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	16	**RCCunit RAM Clock Gating Disable** Access: | R/W RCCunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	15	**PSDunit RAM Clock Gating Disable** Access: | R/W PSDunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)

RCGCTL1 - RAM Clock Gating Control 1

		'1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	14	**MTunit RAM Clock Gating Disable** Access: R/W MTunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	13	**SBEunit RAM Clock gating Disable** Access: R/W SBEunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	12	**IZunit RAM Clock Gating Disable** Access: R/W IZunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	11	**IECPunit RAM Clock Gating Disable** Access: R/W IECPunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	10	**ICunit RAM Clock Gating Disable** Access: R/W ICunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	9	**HIZunit RAM Clock Gating Disable** Access: R/W HIZunit RAM Clock Gating Disable Control:

RCGCTL1 - RAM Clock Gating Control 1

<table>
<tr><td></td><td></td><td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>8</td><td><b>GAMunit RAM Clock Gating Disable</b>
<table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>
GAMunit RAM Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>7</td><td><b>BCunit RAM Clock Gating Disable</b>
<table><tr><td>Access:</td><td>R/W</td></tr></table>
BCunit RAM Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>6</td><td><b>HDCunit RAM Clock Gating Disable</b>
<table><tr><td>Access:</td><td>R/W</td></tr></table>
GAFSunit RAM Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>5</td><td><b>DMunit RAM Clock Gating Disable</b>
<table><tr><td>Access:</td><td>R/W</td></tr></table>
DMunit RAM Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>4</td><td><b>Reserved</b>
<table><tr><td>Access:</td><td>RO</td></tr></table>
Reserved.</td></tr>
<tr><td></td><td>3</td><td><b>CSunit RAM Clock Gating Disable</b>
<table><tr><td>Access:</td><td>R/W</td></tr></table>
CSunit RAM Clock Gating Disable Control:</td></tr>
</table>

RCGCTL1 - RAM Clock Gating Control 1

		'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	2	**BLBunit RAM Clock Gating Disable** Access: R/W BLBunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	1	**MPCunit RAM Clock Gating Disable** Access: R/W MPCunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	0	**BFunit RAM Clock Gating Disable** Access: R/W BFunit RAM Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

RCGCTL2 - RAM Clock Gating Control 2

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	09414h

RAM Clock Gating Control Registers.

DWord	Bit	Description
0	31	**SPARE 2 clock gate disable** Access: R/W SPARE 2 Unit Clock Gating Disable Control: 0: Clock Gating Enabled. (I.e., clocks can be gated when they are not required to toggle for functionality.) 1: Clock Gating Disabled. (I.e., clocks are toggling, always.)
	30:2	**Reserved** Access: RO Reserved.
	1	**VSunit RAM Clock Gating Disable** Access: R/W VSunit RAM Clock Gating Disable Control: 0: Clock Gating Enabled. (I.e., clocks can be gated when they are not required to toggle for functionality.) 1: Clock Gating Disabled. (I.e., clocks are toggling, always.)
	0	**Reserved** Access: RO Reserved.

RCC_LRA_0 - RCC LRA 0

Register Space:	MMIO: 0/2/0
Default Value:	0x3F100F00
Size (in bits):	32
Address:	04058h-0405Bh

RCC LRA 0

DWord	Bit	Description
0	31:30	**Reserved** Default Value: 00b Access: RO Reserved Format: U1
	29:24	**RCC LRA1 Max** Default Value: 111111b Access: R/W RCC LRA1 Max Format: U6 Maximum value of programmable LRA1
	23:22	**Reserved** Default Value: 00b Access: RO Reserved Format: U1
	21:16	**RCC LRA1 Min** Default Value: 010000b Access: R/W RCC LRA1 Min Format: MBZ Minimum value of programmable LRA1
	15:14	**Reserved** Default Value: 00b Access: RO Reserved

RCC_LRA_0 - RCC LRA 0

<table>
<tr><td></td><td></td><td>Format: U1</td></tr>
<tr><td></td><td>13:8</td><td><b>RCC LRA0 Max</b><table><tr><td>Default Value:</td><td>001111b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>RCC LRA0 Max
Format: U1
Maximum value of programmable LRA0</td></tr>
<tr><td></td><td>7:6</td><td><b>Reserved</b><table><tr><td>Default Value:</td><td>00b</td></tr><tr><td>Access:</td><td>RO</td></tr></table>Reserved
Format: U1</td></tr>
<tr><td></td><td>5:0</td><td><b>RCC LRA0 Min</b><table><tr><td>Default Value:</td><td>000000b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>RCC LRA0 Min
Format: U6
Minimum value of programmable LRA0</td></tr>
</table>

RCC_LRA_1 - RCC LRA 1

Register Space:	MMIO: 0/2/0
Default Value:	0x00010000
Size (in bits):	32
Address:	0405Ch-0405Fh

RCC LRA 1

DWord	Bit	Description
0	31:20	**Reserved** Default Value: 000000000000b Access: RO Reserved Format: MBZ
	19:18	**MSC LRA** Default Value: 00b Access: R/W MSC LRA Format: U1 Which LRA should MSC use
	17:16	**RCC LRA** Default Value: 01b Access: R/W RCC LRA Format: MBZ Which LRA should RCC use
	15:0	**Reserved** Default Value: 0000000000000000b Access: RO Reserved Format: MBZ

RCCTLB_VA - RCC Virtual page Address Registers

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04A00h-04A03h

These registers are directly mapped to the current Virtual Addresses in the RCCTLB (Render Cache for Color TLB).

DWord	Bit	Description
0	31:12	**Address** Format: GraphicsAddress[31:12] Page virtual address.
	11:0	**Reserved** Format: MBZ

RCS_BB_STATE - RCS Batch Buffer State Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	02110h

This register contains the attributes of the current batch buffer initiated from the Ring Buffer.

This register should not be written by software. These fields should only get written by a context restore. Software should always set these fields via the MI_BATCH_BUFFER_START command when initiating a batch buffer.
This register is saved and restored with context.

DWord	Bit	Description
0	31:9	**Reserved** Format: MBZ
	8	**Reserved** Format: MBZ
	7	**Reserved** Format: MBZ
	6	**Reserved**
	5	**Address Space Indicator**
	4	**Reserved**
	3:0	**Reserved** Format: MBZ

Bit 5 — Address Space Indicator:

Value	Name	Description
0h	GGTT **[Default]**	This batch buffer is located in GGTT memory
1h	PPGTT	This batch buffer is located in PPGTT memory.

RCZTLB_VA - RCZ Virtual Page Address Registers

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04B00h-04B03h

These registers are directly mapped to the current Virtual Addresses in the RCZTLB (Render Cache for Z (Depth), Hi Z, and Stencil TLB).

DWord	Bit	Description
0	31:12	**Address** Format: GraphicsAddress[31:12] Page virtual address.
	11:0	**Reserved** Format: MBZ

TLBPEND_RDY0 - Ready Bit Vector 0 for TLBPEND registers

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04708h-0470Bh

This register contains the ready bits for entries 0-31 of TLBPEND structure (Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Ready bits per entry**

TLBPEND_RDY1 - Ready Bit Vector 1 for TLBPEND registers

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	0470Ch-0470Fh

This register contains the ready bits for entries 32-63 of TLBPEND structure (Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Ready bits per entry**

RBSYNC - Render/Blitter Semaphore Sync Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	02044h

This register is written by BCS, read by CS.

DWord	Bit	Description
0	31:0	**Semaphore Data** Semaphore data for synchronization between render engine and blitter engine.

MI_MODE - Render Mode Register for Software Interface

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	0209Ch

The MI_MODE register contains information that controls software interface aspects of the Memory Interface function.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Masks</b>
<table><tr><td>Format:</td><td>Mask[15:0]</td></tr></table>
A 1 in a bit in this field allows the modification of the corresponding bit in Bits 15:0</td></tr>
<tr><td></td><td>15</td><td><b>Suspend Flush</b>
<table><tr><td>Format:</td><td>U1</td></tr></table>
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td>No Delay <b>[Default]</b></td><td>HW will not delay flush, this bit will get cleared by MI_SUSPEND_FLUSH as well</td></tr>
<tr><td>1h</td><td>Delay Flush</td><td>Suspend flush is active</td></tr>
</table>
<table>
<tr><th>Programming Notes</th></tr>
<tr><td>This should only be written to from the ring using MI_SUSPEND_FLUSH. It is considered undefined if written by software through MMIO</td></tr>
</table></td></tr>
<tr><td></td><td>14</td><td><b>Async Flip Performance mode</b>
<table><tr><td>Format:</td><td>U1</td></tr></table>
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td>Performance mode enabled <b>[Default]</b></td><td>The stall of the flip event is in the windower</td></tr>
<tr><td>1h</td><td>Performance mode disabled</td><td>The stall of the flip event is in the command stream</td></tr>
</table>
<table>
<tr><th>Programming Notes</th></tr>
<tr><td>This bit must be set to '1' disabling Async Flip Performance mode.</td></tr>
<tr><td>When Async Flip Performance mode is enabled stall is in the Windower allowing the commands following the MI_WAIT_FOR_EVENT to be parsed by command streamer, this breaks the usage model of controlling the display message generation in display engine using MI_LOAD_REGISTER_IMMEDIATE commands from ring buffer.</td></tr>
</table></td></tr>
</table>

MI_MODE - Render Mode Register for Software Interface

13 **Flush Performance mode**

Format:	U1

Value	Name	Description
0h	run fast restore **[Default]**	No NonPipelined SV flush.
1h	run slow legacy restore	With NonPipelined SV flush.

12 **MI_FLUSH Enable**

Format:	Enable

PIPE_CONTROL is a superset of MI_FLUSH. Since MI_FLUSH is redundant, by default, it is disabled

11 **Invalidate UHPTR enable**

Format:	Enable

If bit set H/W clears the valid bit of UHPTR (2134h, bit 0) when current active head pointer is equal to UHPTR.

10 **Reserved**

Format:	MBZ

9 **Rings Idle**

Format:	U1

Read Only Status bit

Value	Name	Description
0h	Not Idle **[Default]**	Parser not Idle or Ring Arbiter not Idle.
1h	Idle	Parser Idle and Ring Arbiter Idle.

Programming Notes
Writes to this bit are not allowed.

8 **Stop Rings**

Format:	U1

Value	Name	Description
0h	**[Default]**	Normal Operation.
1h		Parser is turned off and Ring arbitration is turned off.

Programming Notes
Software must set this bit to force the Rings and Command Parser to Idle. Software must read a 1 in the Ring Idle bit after setting this bit to ensure that the hardware is idle.
Software must clear this bit for Rings to resume normal operation.

7 **Reserved**

Format:	MBZ

MI_MODE - Render Mode Register for Software Interface

<table>
<tr><td></td><td>6</td><td><b>Vertex Shader Timer Dispatch Enable</b>
<table><tr><td>Format:</td><td>Enable</td></tr></table>
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td>Disable <b>[Default]</b></td><td>Disable the timer for dispatch of single vertices from the vertex shader. Vertex shader will try to collect 2 vertices before a dispatch</td></tr>
<tr><td>1h</td><td>Enable</td><td>Enable the timer for dispatch of single vertices. Dispatch a single vertex shader thread after the timer expires.</td></tr>
</table></td></tr>
<tr><td></td><td>5</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>4</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>3:1</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>0</td><td><b>Mask IIR disable</b>
<table><tr><td>Format:</td><td>Disable</td></tr></table>
Mask IIR disable. Nominally the Interrupt controller masks interrupts in the IIR register if an interrupt acknowledge from the 3gio interface is pending. Setting this bit to a 1 allows interrupts to be visible to the interrupt controller while an interrupt acknowledge is pending.</td></tr>
</table>

RVSYNC - Render/Video Semaphore Sync Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	02040h

This register is written by VCS, read by CS.

DWord	Bit	Description
0	31:0	**Semaphore Data** Semaphore data for synchronization between render engine and blitter engine.

PR_CTR - Render Watchdog Counter

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	02190h

DWord	Bit	Description
0	31:0	**Counter Value** Format: U32 This register reflects the render watchdog counter value itself. It cannot be written to.

PR_CTR_THRSH - Render Watchdog Counter Threshold

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00150000
Access:	R/W
Size (in bits):	32
Address:	0217Ch

DWord	Bit	Description
0	31:0	**Counter logic Threshold** Default Value: 00150000h Format: U32 This field specifies the threshold that the hardware checks against for the value of the render clock counter before generating an interrupt. The counter in hardware generates an interrupt when the threshold is reached, rolls over and starts counting again. The interrupt generated is the "Media Hang Notify" interrupt since this watchdog timer is intended primarily to remedy VLD hangs on the main pipeline.

MFC_BITSTREAM_SE_BITCOUNT_FRAME - Reported Bitstream Output Bit Count for Syntax Elements Only Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124A4h

This register stores the count of number of bits in the bitstream due to syntax elements only. This excludes header/ byte alignment /tail/EMU/CABAC-0word/padding bits but includes the stop-one-bit. This register is part of the context save and restore.

DWord	Bit	Description
0	31:0	**MFC Bitstream Syntax Element Only Bit Count** Total number of bits in the bitstream output due to syntax elements only. It includes the data bytes only. This count is updated for every time the internal bitstream counter is incremented and its reset at image start.

MFC_BITSTREAM_BYTECOUNT_FRAME - Reported Bitstream Output Byte Count per Frame Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124A0h

This register stores the count of bytes of the bitstream output per frame

DWord	Bit	Description
0	31:0	**MFC Bitstream Byte Count per Frame** Total number of bytes in the bitstream output per frame from the encoder. This includes header/tail/byte alignment/data bytes/EMU (emulation) bytes/cabac-zero word insertion/padding insertion. This count is updated for every time the internal bitstream counter is incremented and its reset at image start.

MFC_AVC_CABAC_BIN_COUNT_FRAME - Reported Bitstream Output CABAC Bin Count Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	124A8h

This register stores the count of number of bins per frame.

DWord	Bit	Description
0	31:0	**MFC AVC Cabac Bin Count** Total number of BINs in the bitstream output per frame from the encoder. This count is updated for every time the bin counter is incremented and its reset at image start.

TIMESTAMP - Reported Timestamp Count

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	RO. This register is not set by the context restore.
Size (in bits):	64
Address:	02358h

This register provides an elapsed real-time value that can be used as a timestamp for GPU events over short periods of time. Note that the value of this register can be obtained in a 3D pipeline-synchronous fashion without a pipeline flush by using the PIPE_CONTROL command. See 3D Geometry Pipeline in the "3D and Media" volume. This register (effectively) counts at a constant frequency by adjusting the increment amount according to the actual reference clock frequency. SW therefore does not need to know the reference clock frequency. This register is not reset by a graphics reset. It will maintain its value unless a full chipset reset is performed.

DWord	Bit	Description
0	63:36	**Reserved** Format: MBZ
	35:0	**Timestamp Value** Format: U36 **Description** This register toggles every 80 ns. The upper 28 bits are zero.

RSTFCTLMSG - Reset Flow Control Messages

Register Space: MMIO: 0/2/0
Default Value: 0x00000000
Size (in bits): 32

Address: 08108h

Soft-Reset and FLR Flow Control Message Registers

DWord	Bit	Description
0	31:16	**Message Mask** Access: RO Message Mask In order to write to bits 15:0, the corresponding message mask bits must be written. For example, for bit 14 to be set, bit 30 needs to be 1: 40004000
	15:8	**Reserved** Access: RO Reserved
	7	**Blitter Reset Flow Acknowledgement Messages** Access: R/W PM Acknowledgement Messages for Blitter reset: '1': PREP_RST_BLIT_ACK - Acknowledgement that graphics blitter is prepared for reset assertion. '0': DONE_BLIT_RST_ACK - Acknowledgement that graphics blitter reset is de-asserted
	6	**Media Reset Flow Acknowledgement Messages** Access: R/W PM Acknowledgement Messages for Media reset: '1': PREP_RST_MEDIA_ACK - Acknowledgement that graphics media block is prepared for reset assertion. '0': DONE_MEDIA_RST_ACK - Acknowledgement that the graphics media reset is de-asserted
	5	**Render Reset Flow Acknowledgement Messages** Access: R/W PM Acknowledgement Messages for Render reset: '1': PREP_RST_RENDER_ACK - Acknowledgement that the graphics render block is prepared for reset assertion. '0': DONE_RENDER_RST_ACK - Acknowledgement that the graphics render reset is de-asserted

RSTFCTLMSG - Reset Flow Control Messages

	4	**GTI-Device Reset Flow Acknowledgement Messages** Access: R/W PM Acknowledgement Messages for GTI-Device reset: '1': PREP_RST_GTIDEV_ACK - Acknowledgement that the GTI device is prepared for reset assertion. '0': DONE_GTIDEV_RST_ACK - Acknowledgement that the GTI device reset is de-asserted
	3:2	**Reserved** Access: RO Reserved.
	1	**Global Resource Arbitration Acknowledgement Messages** Access: R/W Global Resource Arbitration Acknowledgement Message from PM: '1': CP_ARB_REQ_ACK - Acknowledgement for CPunit's global resource arbitration request '0': CP_ARB_RELEASE_ACK - Acknowledgement to CPunit's release of global resources
	0	**CP Busy / Idle Status Acknowledgement Messages** Access: R/W CP Busy / Idle Status Acknowledgement Message from PM: '0': CP_NOT_BUSY_ACK - Acknowledgement that the CPunit is idle. '1': CP_BUSY_ACK - Acknowledgement that the CPunit is busy.

RING_BUFFER_HEAD_PREEMPT_REG - RING_BUFFER_HEAD_PREEMPT_REG

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32

Address:	0214Ch
Name:	RCS RING_BUFFER_HEAD_PREEMPT_REG
ShortName:	RCS_RING_BUFFER_HEAD_PREEMPT_REG

This register contains the Head pointer offset in the ring when the last PREEMPTABLE command was executed and caused the head pointer to move due to the UHPTR register being valid. If the PREEMPTABLE command is executed as part of the batch buffer then the value of the register will be the offset in the ring of the command past the batch buffer start that contained the preemptable command.
This is a global register and context save/restored as part of power context image.

Programming Notes
Programming Restriction: **This register should NEVER be programmed by driver. This is for HW internal use only.**

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:21</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>20:2</td><td><b>Preempted Head Offset</b><table><tr><td>Format:</td><td>U19</td></tr></table>This field contains the Head pointer offset in the ring when the last MI_ARB_CHECK command was executed and caused the head pointer to move due to the UHPTR register being valid.</td></tr>
<tr><td></td><td>1:0</td><td><b>Ring/Batch Indicator</b><table><tr><td>Format:</td><td>Enabled</td></tr></table><table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td>Ring</td><td>Preemptable command was executed in ring and caused head pointer to be updated.</td></tr><tr><td>1h</td><td>Batch</td><td>Preemptable command was executed in batch and caused head pointer to be updated.</td></tr></table></td></tr>
</table>

RING_BUFFER_CTL - Ring Buffer Control

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32

Address:	0203Ch
Name:	RCS Ring Buffer Control
ShortName:	RCS_RING_BUFFER_CTL

Address:	1203Ch
Name:	VCS Ring Buffer Control
ShortName:	VCS_RING_BUFFER_CTL

Address:	2203Ch
Name:	BCS Ring Buffer Control
ShortName:	BCS_RING_BUFFER_CTL

Description
These registers are used to define and operate the ring buffer mechanism which can be used to pass instructions to the command interface. The buffer itself is located in a physical memory region. The ring buffer is defined by a 4 Dword register set that includes starting address, length, head offset, tail offset, and control information. Refer to the Programming Interface chapter for a detailed description of the parameters specified in this ring buffer register set, restrictions on the placement of ring buffer memory, arbitration rules, and in how the ring buffer can be used to pass instructions.
Ring Buffer Head and Tail Offsets must be properly programmed before it is enabled. A Ring Buffer can be enabled when empty.
Graphics Engine doesn't go IDLE when head offset is not equal to tail offset when ring buffer is disabled. PSMI controller waits for HW to go Idle as part of the PSMI flow. When PSMI flow happens in middle of ring buffer initialization where in Head offset is not equal to Tail offset and Ring Buffer disabled, PMSI flow will hang waiting for Graphics Engine to go IDLE. (During ring buffer initialization SW programs Head and Tail offsets prior to enabling Ring Buffer). In order to avoid this dead lock PSMI controller must detect this case and program head and tail offset to be equal to allow Graphics Engine to go IDLE, before exiting PSMI flow original head and tail offsets should be restored.

DWord	Bit	Description
0	31:21	**Reserved** Format: MBZ
	20:12	**Buffer Length** Format: U9-1 in 4 KB pages - 1 This field is written by SW to specify the length of the ring buffer in 4 KB Pages. Range = [0 = 1 page = 4 KB, 1FFh = 512 pages = 2 MB]

Value	Name	Description
0		1 page = 4 KB

RING_BUFFER_CTL - Ring Buffer Control

1FFh		512 pages = 2 MB

11 **RBWait**

Description
Indicates that this ring has executed a WAIT_FOR_EVENT instruction and is currently waiting. Software can write a "1" to clear this bit, write of "0" has no effect. When the RB is waiting for an event and this bit is cleared, the wait will be terminated and the RB will be returned to arbitration.
RenderCS: RBWait is not set on executing WAIIT_FOR_EVENT instruction waiting on Async Flip Pending.

10 **Semaphore Wait**

Description
Indicates that this ring has executed a MI_SEMAPHORE_MBOX instruction with register compare and is currently waiting.

9 **Reserved**

Format:	MBZ

8 **Reserved**

Format:	MBZ

7:3 **Reserved**

Format:	MBZ

2:1 **Automatic Report Head Pointer**

Source:	BlitterCS, VideoCS
Exists If:	//BCS, VCS

Description
This field is written by software to control the automatic "reporting" (write) of this ring buffer's "Head Pointer" register (register DWord 1) to the corresponding location within the Hardware Status Page. Automatic reporting can either be disabled or enabled at 4KB, 64KB or 128KB boundaries within the ring buffer.

Value	Name	Description
0	MI_AUTOREPORT_OFF	Automatic reporting disabled
1	MI_AUTOREPORT_64KB	Report every 16 pages (64KB)
2	MI_AUTOREPORT_4KB	Report every page (4KB) This mode must not be enabled in Ring Buffer mode of scheduling to minimize the auto reports.
3	MI_AUTOREPORT_128KB	Report every 32 pages (128KB)

2:1 **Automatic Report Head Pointer**

Source:	RenderCS

RING_BUFFER_CTL - Ring Buffer Control

Exists If:	//RCS

This field is written by software to control the automatic "reporting" (write) of this ring buffer's "Head Pointer" register (register DWord 1) to the corresponding location within the Hardware Status Page. Automatic reporting can either be disabled or enabled at 4KB, 64KB or 128KB boundaries within the ring buffer.

Value	Name	Description
0h	MI_AUTOREPORT_OFF	Automatic reporting disabled
1h	Reserved	Reserved
2h	Reserved	
3h	MI_AUTOREPORT_128KB	Report every 32 pages (128KB)

Programming Notes

0

Ring Buffer Enable

Format:	Enable

This field is used to enable or disable this ring buffer. It can be enabled or disabled regardless of whether there are valid instructions pending. If disabled and the ring head equals ring tail, all state currently loaded in hardware is considered invalid.

Programming Notes
Ring Buffer Mode of Scheduling: SW must follow the below programming notes during SW initialization phase or while enabling render engine's ring buffer for the first time, this would be coming out of boot, standby, hibernate or reset. • SW must set the Force Wakeup bit to prevent GT from entering C6. • SW must dispatch workload (dummy context) to initialize render engine with default state such that any context switches that occur subsequently (Power Save) will save and restore coherent device state. Indirect pointers used in 3D states must point to valid graphics surface existing in memory. PP_DCLV followed by PP_DIR_BASE register should be programmed as part of initialization workload if PPGTT is enabled in GFX_MODE register. • SW must ensure all the register (MMIO) initialization/programming through CPU happens in this block or latter, this ensures the MMIO state is save/restored on subsequent context switches (Power Sequences). • Once the render engine is programmed with valid state and the configuration, Force Wakeup bit should be reset to enable C6 entry.

BCS_RCCID - Ring Buffer Current Context ID Register

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Address:	22190h-22197h

This register contains the current ring context ID associated with the ring buffer.

Programming Notes
The current context registers must not be written directly (via MMIO). The RCCID register should only be updated indirectly from RNCID.

DWord	Bit	Description
0	63:0	**Unnamed** See Context Descriptor for BCS.

RING_BUFFER_HEAD - Ring Buffer Head

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02034h
Name:	RCS Ring Buffer Head
ShortName:	RCS_RING_BUFFER_HEAD
Address:	12034h
Name:	VCS Ring Buffer Head
ShortName:	VCS_RING_BUFFER_HEAD
Address:	22034h
Name:	BCS Ring Buffer Head
ShortName:	BCS_RING_BUFFER_HEAD

This register is used to define and operate the ring buffer mechanism which can be used to pass instructions to the command interface. The buffer itself is located in a physical memory region. The ring buffer is defined by a 4 Dword register set that includes starting address, length, head offset, tail offset, and control information. Refer to the Programming Interface chapter for a detailed description of the parameters specified in this ring buffer register set, restrictions on the placement of ring buffer memory, arbitration rules, and in how the ring buffer can be used to pass instructions.

Ring Buffer Head Offsets must be properly programmed before ring is enabled. A Ring Buffer can be enabled when empty.

DWord	Bit	Description
0	31:21	**Wrap Count** Format: U11 count of ring buffer wraps This field is incremented by 1 whenever the **Head Offset** wraps from the end of the buffer back to the start (i.e., whenever it wraps back to 0). Appending this field to the **Head Offset** field effectively creates a virtual 4GB Head "Pointer" which can be used as a tag associated with instructions placed in a ring buffer. The Wrap Count itself will wrap to 0 upon overflow.
	20:2	**Head Offset** Format: GraphicsAddress[20:2] DWord Offset This field indicates the offset of the *next* instruction DWord to be parsed. Software will initialize this field to select the first DWord to be parsed once the RB is enabled. (Writing the Head Offset while the RB is enabled is UNDEFINED). Subsequently, the device will increment this offset as it executes instructions - until it reaches the QWord specified by the **Tail Offset**. At this point the ring buffer is considered "empty". **Programming Notes** A RB can be enabled empty or containing some number of valid instructions.
	1	**Reserved**

RING_BUFFER_HEAD - Ring Buffer Head

<table>
<tr><td></td><td></td><td><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>0</td><td><b>Wait for Condition Indicator</b><table><tr><td>Source:</td><td>RenderCS</td></tr><tr><td>Exists If:</td><td>//RCS</td></tr></table>This is a read only value used to indicate whether or not the command streamer is currently waiting for a conditional code to be cleared from 0x2028</td></tr>
<tr><td></td><td>0</td><td><b>Reserved</b><table><tr><td>Source:</td><td>BlitterCS, VideoCS</td></tr><tr><td>Exists If:</td><td>//BCS, VCS</td></tr><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
</table>

RING_BUFFER_START - Ring Buffer Start

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02038h
Name:	RCS Ring Buffer Start
ShortName:	RCS_RING_BUFFER_START
Address:	12038h
Name:	VCS Ring Buffer Start
ShortName:	VCS_RING_BUFFER_START
Address:	22038h
Name:	BCS Ring Buffer Start
ShortName:	BCS_RING_BUFFER_START

These registers are used to define and operate the "ring buffer" mechanism which can be used to pass instructions to the command interface. The buffer itself is located in a physical memory region. The ring buffer is defined by a 4 Dword register set that includes starting address, length, head offset, tail offset, and control information. Refer to the Programming Interface chapter for a detailed description of the parameters specified in this ring buffer register set, restrictions on the placement of ring buffer memory, arbitration rules, and in how the ring buffer can be used to pass instructions.

DWord	Bit	Description
0	31:12	**Starting Address** Format: GraphicsAddress[31:12]RingBuffer This field specifies Bits 31:12 of the 4KB-aligned starting Graphics Address of the ring buffer. Address bits 31 down to 29 must be zero. All ring buffer pages must map to Main Memory (uncached) pages. Ring Buffer addresses are always translated through the global GTT.
	11:0	**Reserved** Format: MBZ

RING_BUFFER_TAIL - Ring Buffer Tail

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32

Address:	02030h
Name:	RCS Ring Buffer Tail
ShortName:	RCS_RING_BUFFER_TAIL

Address:	12030h
Name:	VCS Ring Buffer Tail
ShortName:	VCS_RING_BUFFER_TAIL

Address:	22030h
Name:	BCS Ring Buffer Tail
ShortName:	BCS_RING_BUFFER_TAIL

These registers are used to define and operate the "ring buffer" mechanism which can be used to pass instructions to the command interface. The buffer itself is located in a linear memory region. The ring buffer is defined by a 4 Dword register set that includes starting address, length, head offset, tail offset, and control information.

Refer to the Programming Interface chapter for a detailed description of the parameters specified in this ring buffer register set, restrictions on the placement of ring buffer memory, arbitration rules, and in how the ring buffer can be used to pass instructions.

Ring Buffer Tail Offsets must be properly programmed before ring is enabled. A Ring Buffer can be enabled when empty.

DWord	Bit	Description
0	31:21	**Reserved** Format: MBZ
	20:3	**Tail Offset** Format: GraphicsAddress[20:3] This field is written by software to specify where the valid instructions placed in the ring buffer end. The value written points to the QWord past the last valid QWord of instructions. In other words, it can be defined as the next QWord that software will write instructions into. Software must write subsequent instructions to QWords following the Tail Offset, possibly wrapping around to the top of the buffer (i.e., software can't skip around within the buffer). Note that all DWords prior to the location indicated by the **Tail Offset** must contain valid instruction data - which may require instruction padding by software. See **Head Offset** for more information.
	2:0	**Reserved** Format: MBZ

TLBPEND_SEC0 - Section 0 of TLBPEND Entry

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04400h-04403h

This register is directly mapped to the TLBPEND Array in the Graphic Arbiter.

DWord	Bit	Description
0	31	**vtstatus** This bit will be used in conjunction with the ready bit to determine the stage of the translation. See table below.
	30:28	**GTT bits** Bits 3:1 of the GTT entry used to translate the Virtual Address. 000 if translation is pending.
	27:0	**Current address** The value of this field depends on the stage of the TLB translation for this entry: VA - bits 27:20 = 00, bits 19:0 = Bits 31:12 of the Virtual Address of the cycle.

TLBPEND_SEC1 - Section 1 of TLBPEND Entry

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04500h-04503h

This register is directly mapped to the current Virtual Addresses in the MTTLB (Texture and constant cache TLBRender Cache for Z (Depth), Hi Z, and Stencil TLB).

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:28</td><td><b>Current address</b>
Bits 9:6 of the Virtual Address of the cycle.</td></tr>
<tr><td></td><td>27:24</td><td><b>Cacheability Control Bits</b>
<table>
<tr><td>Bits 3:1 of the GTT entry are used to translate the Virtual Address. 000 if translation is pending.</td></tr>
<tr><td>3 <b>Reserved</b></td></tr>
<tr><td>2 <b>Graphics Data Type (GFDT).</b> This field contains the GFDT bit for this surface when writes occur. GFDT can also be set by the GTT. The effective GFDT is the logical OR of this field with the GFDT from the GTT entry. This field is ignored for reads.
1:0 <b>Cacheability Control.</b> This field controls cacheability.
00: Use cacheability control bits from GTT entry.
01: Data is not cached.
11: Data is cached.</td></tr>
</table></td></tr>
<tr><td></td><td>23</td><td><b>ZLR bit</b>
Flag to indicate this is a zero length read, a read used to calculate a physical address for a write.</td></tr>
<tr><td></td><td>22:4</td><td><b>TAG</b>
Cycle identification TAG.</td></tr>
<tr><td></td><td>3:0</td><td><b>SRC ID</b>
Encoding of unit generating this cycle.
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0000b</td><td>CS_RD_SRCID</td></tr>
<tr><td>0001b</td><td>VF_RD_SRCID</td></tr>
<tr><td>0010b</td><td>ISC_SRCID</td></tr>
<tr><td>0011b</td><td>MT_SRCID</td></tr>
<tr><td>0100b</td><td>RCC_SRCID</td></tr>
<tr><td>0101b</td><td>HZARB_SRCID</td></tr>
<tr><td>0110b</td><td>RCZ_SRCID</td></tr>
<tr><td>0111b</td><td>CTC_SRCID</td></tr>
</table></td></tr>
</table>

TLBPEND_SEC1 - Section 1 of TLBPEND Entry

		1000b	CS_WR_SRCID
		1001b	MBC_SRCID
		1010b	CS_RD_PROBE
		1011b	CS_RD_PWRCTX
		1100b	RC_R4WRCMP
		1101b	RESRVD2_SRCID
		1110b	RESRVD1_SRCID
		1111b	RESRVD0_SRCID

TLBPEND_SEC2 - Section 2 of TLBPEND Entry

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04600h-04603h

This register is directly mapped to the current Virtual Addresses in the MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:14	**Reserved**
	13	**Big Page Attribute** This entry is using a 32K page.
	12:8	**Current Address** Format: GraphicsAddress[14:10] Bits 14:10 of the Virtual Address of the cycle.
	7:0	**PAT Entry** Location of Physical Address in Physical Address Table.

SO_NUM_PRIMS_WRITTEN[0:3] - Stream Output Num Primitives Written Counter

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Address:	05200h-0521Fh

There is one 64-bit register for each of the 4 supported streams:5200h-5207h SO_NUM_PRIMS_WRITTEN0 (for Stream Out Stream #0)5208h-520Fh SO_NUM_PRIMS_WRITTEN1 (for Stream Out Stream #1)5210h-5217h SO_NUM_PRIMS_WRITTEN2 (for Stream Out Stream #2)5218h-521Fh SO_NUM_PRIMS_WRITTEN3 (for Stream Out Stream #3)These registers are used to count the number of primitives (aka objects: points, lines, triangles) which the SO stage has successfully written to a particular "stream's" Streamed Vertex Output buffers, subject to buffer overflow detection. (See the Stream Output section of the 3D pipeline volume). These registers are part of the context save and restore.

DWord	Bit	Description
0	63:0	**Num Prims Written Count** Format: U64 This count is incremented (by one) every time a GS thread outputs a DataPort Streamed Vertex Buffer Write message with the Increment Num Prims Written bit set in the message header (see the Geometry Shader and Data Port chapters in the 3D Volume.)

SO_PRIM_STORAGE_NEEDED[0:3] - Stream Output Primitive Storage Needed Counters

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	RO. This register is set by the context restore.
Size (in bits):	64
Address:	05240h-0525Fh

There is one 64-bit register for each of the 4 supported streams:

5240h-5247h SO_PRIM_STORAGE_NEEDED0 (for Stream Out Stream #0)
5248h-524Fh SO_PRIM_STORAGE_NEEDED1 (for Stream Out Stream #1)
5250h-5257h SO_PRIM_STORAGE_NEEDED2 (for Stream Out Stream #2)
5258h-525Fh SO_PRIM_STORAGE_NEEDED3 (for Stream Out Stream #3)

These registers are used to count the number of primitives (aka objects: points, lines, triangles) which the SO stage has or would have written to a particular "stream's" Streamed Vertex Output buffers if all buffers had been large enough to accommodate the writes. (See the Stream Output section of the 3D pipeline volume).

These registers are part of the context save and restore.

DWord	Bit	Description
0	63:0	**Prim Storage Needed Count** Format: U64 This count is incremented (by one) by the SOL stage for each object (point, line, triangle) it writes or attempts to write to the corresponding stream's output buffers. The count is not affected by the actual number of buffers bound to the stream.

SO_WRITE_OFFSET[0:3] - Stream Output Write Offsets

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RW. This register is set by the context restore.
Size (in bits):	32
Address:	05280h-0528Fh

There is one R/W 32-bit register for each of the 4 supported stream output buffer slots:
5280h-5283h SO_WRITE_OFFSET0 (for Stream Out Buffer #0)
5284h-5287h SO_ WRITE_OFFSET1 (for Stream Out Buffer #1)
5288h-528Bh SO_ WRITE_OFFSET2 (for Stream Out Buffer #2)
528Ch-528Fh SO_ WRITE_OFFSET3 (for Stream Out Buffer #3)
These registers are used to set and track a DWord-granular Write Offset for each of the 4 Stream Output Buffer slots. Software can directly write them via MI_LOAD_REGxxx commands. The SOL stage will increment them as part of stream output processing. Software can cause them to be written to memory via MI_STORE_REGxxx commands. (See the Stream Output section of the 3D pipeline volume).
These registers are part of the context save and restore.

Programming Notes

- Software must ensure that no HW stream output operations can be in process or otherwise pending at the point that the MI_LOAD/STORE commands are processed. This will likely require a pipeline flush.
- The SOL stage will effectively advance the write offset by the buffer's Surface Pitch after each vertex is written (assuming no overflow is detected in any targetted SO buffer). Under "normal" conditions one would expect software to initialize the WriteOffset to some (possibly zero) multiple of Surface Pitch in order to align vertex writes to the buffer's Base Address, though it is not required to do so.

DWord	Bit	Description
0	31:2	**Write Offset** Format: U30 This field contains a DWord offset from the corresponding SO buffer's Base Address value. The SOL stage uses this value as a write offset when performing writes to the buffer. The SOL stage will increment this value as a part of performing stream output to the buffer. Note that the SOL stage uses the buffer's Surface Pitch to advance the Write Offset, without regard to the buffer's Base Address (see Programming Notes above).
	1:0	**Reserved** Format: MBZ

FF_MODE - Thread Mode Register

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x28A01010
Access:	R/W
Size (in bits):	32
Address:	020A0h

This register is used to program the FF shader Mode.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table>
</td></tr>
<tr><td></td><td>30</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table>
</td></tr>
<tr><td></td><td>29:26</td><td><b>DS Hit Max Value</b>
<table><tr><td>Format:</td><td>U4</td></tr></table>
<table><tr><th>Description</th></tr>
<tr><td>If the number of hits reaches the DS Hit Max Value and there is a pending miss to be dispatched, the DS will dispatch the pending miss vertex as a single dispatch.</td></tr>
<tr><td>Programming the value beyond the range will have undefined behavior.</td></tr></table>
<table><tr><th>Value</th><th>Name</th></tr>
<tr><td>10</td><td><b>[Default]</b></td></tr>
<tr><td>[1,11]</td><td></td></tr></table>
</td></tr>
<tr><td></td><td>25:20</td><td><b>VS Hit Max Value</b>
<table><tr><td>Format:</td><td>U6</td></tr></table>
<table><tr><th>Description</th></tr>
<tr><td>If the number of hits reaches the VS Hit Max Value and there is a pending miss to be dispatched, the VS will dispatch the pending miss vertex as a single dispatch.</td></tr>
<tr><td>Programming the value beyond the range will have undefined behavior.</td></tr></table>
<table><tr><th>Value</th><th>Name</th></tr>
<tr><td>10</td><td><b>[Default]</b></td></tr>
<tr><td>[1,58]</td><td></td></tr></table>
</td></tr>
<tr><td></td><td>19</td><td><b>DS Reference Count Full Force Miss Enable</b>
<table><tr><td>Format:</td><td>Enable</td></tr></table>
<table><tr><th>Value</th><th>Name</th><th>Description</th></tr></table>
</td></tr>
</table>

FF_MODE - Thread Mode Register

0b	**[Default]**	On a hit to the DS cache and the associated handle's reference count is full then stall until a derefernce.
1b		On a hit to the DS cache and the associated handle's reference count is full then force the cycle as a miss and allocate a new handle.

18:17 **TS Thread Dispatch Mode**

Format:	U2

Value	Name	Description
0h	Load Balanced **[Default]**	Thread Dispatch will load balance the half slices of the threads. Note: this will cause possible corruption if input handles are reused due to instancing or topologies that reuse vertices(i.e. strips and fans)
1h	Half Slice 0	All threads will be dispatched to Half Slice 0.
2h	Half Slice 1	All threads will be dispatched to Half Slice 1.
3h	Reserved	

16 **TS Thread Dispatch Override Enable**

Format:	Enable

Value	Name	Description
0h	Disable **[Default]**	Hardware will decide which half slice the thread will dispatch.
1h	Enable	The value in the TS Thread Dispatch Mode will be used for dispatch.

15 **VS Reference Count Full Force Miss Enable**

Format:	U1

Value	Name	Description
[0,1]		
0b	**[Default]**	On a hit to the VS cache and the associated handle's reference count is full then stall until a derefernce.
1b		On a hit to the VS cache and the associated handle's reference count is full then force the cycle as a miss and allocate a new handle.

14:13 **VS Thread Dispatch Mode**

Format:	U2

Value	Name	Description
0h	Load Balanced **[Default]**	Thread Dispatch will load balance the half slices of the threads. Note: this will cause possible corruption if input handles are reused due to instancing or topologies that reuse vertices (i.e. strips and fans)
1h	Half Slice 0	All threads will be dispatched to Half Slice 0.
2h	Half Slice 1	All threads will be dispatched to Half Slice 1.
3h	Reserved	

FF_MODE - Thread Mode Register

12 **VS Thread Dispatch Override Enable**

Format:	Enable

Value	Name	Description
0h	Disable	Hardware will decide which half slice the thread will dispatch.
1h	Enable **[Default]**	The value in the VS Thread Dispatch Mode will be used for dispatch.

11:7 **Reserved**

Format:	MBZ

6:5 **DS Thread Dispatch Mode**

Format:	U2

Value	Name	Description
0h	Load Balanced **[Default]**	Thread Dispatch will load balance the half slices of the threads. Note: this will cause possible corruption if input handles are reused due to instancing or topologies that reuse vertices (i.e., strips and fans).
1h	Half Slice 0	All threads will be dispatched to Half Slice 0.
2h	Half Slice 1	All threads will be dispatched to Half Slice 1.
3h	Reserved	

4 **DS Thread Dispatch Override Enable**

Format:	Enable

Value	Name	Description
0h	Disable	Hardware will decide which half slice the thread will dispatch.
1h	Enable **[Default]**	The value in the DS Thread Dispatch Mode will be used for dispatch.

3:0 **Reserved**

Format:	MBZ

TLB_RD_ADDR - TLB_RD_ADDRESS Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04700h-04703h

TLB Read Address

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:10</td><td><b>Reserved</b>
<table>
<tr><td>Default Value:</td><td>0000000000000000000000b</td></tr>
<tr><td>Access:</td><td>RO</td></tr>
</table>
Reserved Bits</td></tr>
<tr><td></td><td>9:0</td><td><b>TLB Read Address</b>
<table>
<tr><td>Default Value:</td><td>0000000000b</td></tr>
<tr><td>Access:</td><td>R/W</td></tr>
</table>
TLB Read Address
MSB<9:X> :
<table>
<tr><td>TLB Select</td><td><9:X></td><td>PAT MSB: Section of the PAT used.</td></tr>
<tr><td>PAT_MSB_VLFTLB</td><td>00000</td><td>32 entries - 32</td></tr>
<tr><td>PAT_MSB_CVSTLB</td><td>00001</td><td>32 entries - 32</td></tr>
<tr><td>PAT_MSB_RCCTLB</td><td>0001</td><td>64 entries - 64</td></tr>
<tr><td>PAT_MSB_ZTLB</td><td>001</td><td>128 entries - 128</td></tr>
<tr><td>PAT_MSB_L3TLB</td><td>01</td><td>160 entries - 256</td></tr>
<tr><td>PAT_MSB_CASCTLB</td><td>10</td><td>140 entries - 256</td></tr>
</table>
LSB <X:0> :
GEN RAM ADDRES in Selected TLB</td></tr>
</table>

TLB_RD_DATA - TLB_RD_DATA Register

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04704h-04707h

TLB_READ_DATA Register

DWord	Bit	Description
0	31:0	**TLB_READ_DATA Register** Default Value: 00000000h Access: RO Return data

TLB064_VA - TLB064_VA Virtual Page Address Registers

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	14800h-14803h

This register is directly mapped to the current Virtual Addresses in the TLB064 (VCS and VMC TLB).

DWord	Bit	Description
0	31:12	**Address** Format: GraphicsAddress[31:12] Page virtual address.
	11:0	**Reserved** Format: MBZ

TLB132_VA - TLB132_VA Virtual Page Address Registers

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	14900h-14903h

These registers are directly mapped to the current Virtual Addresses in the TLB132 (All The Media Clients TLB). Default Value = UUUUUUUUh Trusted Type = 1

DWord	Bit	Description
0	31:12	**Address** Format: GraphicsAddress[31:12] Page virtual address.
	11:0	**Reserved** Format: MBZ

TLB232_VA - TLB232_VA Virtual Page Address Registers

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	14A00h-14A03h

This register is directly mapped to the current Virtual Addresses in the TLB232 (VDS and VLF FW TLB).

DWord	Bit	Description
0	31:12	**Address** Format: GraphicsAddress[31:12] Page virtual address.
	11:0	**Reserved** Format: MBZ

TLB304_VA - TLB304_VA Virtual Page Address Registers

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	14B00h-14B03h

This register is directly mapped to the current Virtual Addresses in the TLB304 (VCR TLB).

DWord	Bit	Description
0	31:12	**Address** Format: GraphicsAddress[31:12] Page virtual address.
	11:0	**Reserved** Format: MBZ

GFX_PEND_TLB - TLBPEND Control Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	14040h

Max Outstanding Media pending TLB requests

DWord	Bit	Description
0	31	**Reserved** Format: MBZ
	30	**Reserved** Format: MBZ
	29:24	**VMX BS Limit Count** Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	23	**VMC Limit Enable bit** Format: U1 This bit is used to enable the pending TLB requests limitation function for the VMC. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	22	**Reserved** Format: MBZ
	21:16	**VMC TLB Limit Count** Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	15	**VMXRS Limit Enable bit** Format: U1 This bit is used to enable the pending TLB requests limitation function for the VMX Row store. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	14	**Reserved**

GFX_PEND_TLB - TLBPEND Control Register

		Format: MBZ
	13:8	**VMX RS Random Accsess TLB Limit Count** Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.
	7	**VCS Limit Enable bit** Format: U1 This bit is used to enable the pending TLB requests limitation function for the Command Streamer. When set, the number of internal pending read requests which require a TLB read will not exceed the programmed counter value.
	6	**Reserved** Format: MBZ
	5:0	**VCS TLB Limit Count** Format: U6 This is the MAX number of Allowed internal pending read requests which require a TLB read.

UCGCTL1 - Unit Level Clock Gating Control 1

Register Space: MMIO: 0/2/0
Default Value: 0x02800000
Size (in bits): 32

Address: 09400h

Unit Level Clock Gating Control Registers.

DWord	Bit	Description
0	31	**Sarbunit Clock Gating Disable** Access: R/W SARB unit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	30	**IEFunit Clock Gating Disable** Access: R/W IEFunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	29	**IECPunit Clock Gating Disable** Access: R/W IECPunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	28	**ICunit Clock Gating Disable** Access: R/W ICunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	27	**HIZunit Clock Gating Disable** Access: R/W HIZunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

UCGCTL1 - Unit Level Clock Gating Control 1

	26	**GWunit Clock Gating Disable** Access: | R/W GWunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	25	**GTIunit Clock Gating Disable** Default Value: | 1b Access: | R/W GTI Units Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	24	**GSunit Clock Gating Disable** Access: | R/W GSunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	23	**GPMunit Clock Gating Disable** Default Value: | 1b Access: | R/W GPMunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	22	**GAMunit Clock Gating Disable** Default Value: | 0b Access: | R/W GAMunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	21	**GACunit Clock Gating Disable**

UCGCTL1 - Unit Level Clock Gating Control 1

		Default Value: 0b Access: R/W GACunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	20	**GABunit Clock Gating Disable** Default Value: 0b Access: R/W GABunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	19	**FTunit Clock Gating Disable** Access: R/W FTunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	18	**FLunit Clock Gating Disable** Access: R/W FLunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	17	**EU_FPUunit Clock Gating Disable** Access: R/W EU_FPUunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	16	**EU_TCunit Clock Gating Disable** Access: R/W EU_TCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle

UCGCTL1 - Unit Level Clock Gating Control 1

		for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	15	**EU_EMunit Clock Gating Disable** Access: R/W EU_EMunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	14	**EU_GAunit Clock Gating Disable** Access: R/W EU_GAunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	13	**EUunit Clock Gating Disable** Access: R/W EUunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	12	**SVLunit Clock Gating Disable** Access: R/W SVLunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	11	**DTunit Clock Gating Disable** Access: R/W DTunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	10	**DMunit Clock Gating Disable** Access: R/W

UCGCTL1 - Unit Level Clock Gating Control 1

		DMunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	9	**DGunit Clock Gating Disable** Access: R/W DGunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	8	**DAPunit Clock Gating Disable** Access: R/W DAPunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	7	**CSunit Clock Gating Disable** Access: R/W CSunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	6	**CLunit Clock Gating Disable** Access: R/W CLunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	5	**BLBunit Clock Gating Disable** Access: R/W BLBunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	4	**BFunit Clock Gating Disable**

UCGCTL1 - Unit Level Clock Gating Control 1

		Access: R/W BFunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	3	**BDunit Clock Gating Disable** Access: R/W BDunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	2	**BCSunit Clock Gating Disable** Access: R/W BCSunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	1	**AVSunit Clock Gating Disable** Access: R/W AVSunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	0	**Reserved** Access: RO Reserved.

UCGCTL2 - Unit Level Clock Gating Control 2

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	09404h

Unit Level Clock Gating Control Registers.

DWord	Bit	Description
0	31	**VFunit Clock Gating Disable** Access: R/W VFunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	30	**VDSunit Clock Gating Disable** Access: R/W VDSunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	29	**VDIunit Clock Gating Disable** Access: R/W VDIunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	28	**VCSunit Clock Gating Disable** Access: R/W VCSunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	27	**DTOunit Clock Gating Disable** Access: R/W DTOunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

UCGCTL2 - Unit Level Clock Gating Control 2

	26	**VCPunit Clock Gating Disable** Access: R/W VCPunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	25	**VCDunit Clock Gating Disable** Access: R/W VCDunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	24	**URBMunit Clock Gating Disable** Access: R/W URBMunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	23	**TSGunit Clock Gating Disable** Access: R/W TSGunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	22	**TDLunit Clock Gating Disable** Access: R/W TDLunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	21	**TDSunit Clock Gating Disable** Access: R/W TDSunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle

UCGCTL2 - Unit Level Clock Gating Control 2

		for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	20	**SVSMunit Clock Gating Disable** Access: R/W SVSMunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	19	**SVGunit Clock Gating Disable** Access: R/W SVGunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	18	**SOunit Clock Gating Disable** Access: R/W SOunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	17	**SIunit Clock Gating Disable** Access: R/W SIunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	16	**SFunit Clock Gating Disable** Access: R/W SFunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	15	**SECunit Clock Gating Disable** Access: R/W

UCGCTL2 - Unit Level Clock Gating Control 2

		SECunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	14	**SCunit Clock Gating Disable** Access: | R/W SCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	13	**RCZunit Clock Gating Disable** Access: | R/W RCZunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	12	**RCPBunit Clock Gating Disable** Access: | R/W RCPBunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	11	**RCCunit Clock Gating Disable** Access: | R/W RCCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	10	**QCunit Clock Gating Disable** Access: | R/W QCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	9	**PSDunit Clock Gating Disable**

UCGCTL2 - Unit Level Clock Gating Control 2

		Access: R/W PSDunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	8	**PLunit Clock Gating Disable** Access: R/W PLunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	7	**MTunit Clock Gating Disable** Access: R/W MTunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	6	**MPCunit Clock Gating Disable** Access: R/W MPCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	5	**TDGunitClock Gating Disable** Access: R/W TDGunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	4	**MSCunit Clock Gating Disable** Access: R/W MSCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

UCGCTL2 - Unit Level Clock Gating Control 2

	3	**TEunit Clock Gating Disable** Access: R/W TEunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	2	**TETGunit Clock Gating Disable** Access: R/W TETGunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	1	**MAunit Clock Gating Disable** Access: R/W MAunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	0	**IZunit Clock Gating Disable** Access: R/W IZunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

UCGCTL3 - Unit Level Clock Gating Control 3

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	09408h

Unit Level Clock Gating Control Registers.

DWord	Bit	Description
0	31	**Flunits 2nd Clock Gating Disable** Access: R/W Flunits 2nd Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	30	**SVRRunit Clock Gating Disable** Access: R/W SVRRunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	29	**VCRunit Clock Gating Disable** Access: R/W VCRunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	28	**EDTunit Clock Gating Disable** Access: R/W EDTunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	27	**VCIunit Clock Gating Disable** Access: R/W VCIunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

UCGCTL3 - Unit Level Clock Gating Control 3

	26	**Reserved** Access: RO Reserved.
	25	**HSunit Clock Gating Disable** Access: R/W HSunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	24	**SOLunit Clock Gating Disable** Access: R/W SOLunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	23	**QRCunit Clock Gating Disable** Access: R/W QRCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	22	**MSPBISTunit Clock Gating Disable** Access: R/W MSPBISTunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	21	**BSPunit Clock Gating Disable** Access: R/W BSPunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

UCGCTL3 - Unit Level Clock Gating Control 3

	20	**Reserved**
	19	**SBEunit Clock Gating Disable** Access: R/W SBEunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	18	**BCunit Clock Gating Disable** Access: R/W BCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	17	**WMBE Clock Gating Disable** Access: R/W WMBEunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	16	**WMFEunit Clock Gating Disable** Access: R/W WMFEunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	15	**VSCunit Clock Gating Disable** Access: R/W VSCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	14	**Reserved**
	13	**USBunit Clock Gating Disable** Access: R/W USBunit Clock Gating Disable Control:

UCGCTL3 - Unit Level Clock Gating Control 3

		'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	12	**STCunit Clock Gating Disable** Access: | R/W STCunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	11	**VSunit Clock Gating Disable** Access: | R/W VSunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	10	**VOPunit Clock Gating Disable** Access: | R/W VOPunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	9	**VMXunit Clock Gating Disable** Access: | R/W VMXunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	8	**VMEunit Clock Gating Disable** Access: | R/W VMEunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	7	**VMDunit Clock Gating Disable**

UCGCTL3 - Unit Level Clock Gating Control 3

<table>
<tr><td></td><td></td><td><table><tr><td>Access:</td><td>R/W</td></tr></table>VMDunit Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>6</td><td><b>VMCunit Clock Gating Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table>VMCunit Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>5</td><td><b>VLFunit Clock Gating Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table>VLFunit Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>4</td><td><b>VITunit Clock Gating Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table>VITunit Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>3</td><td><b>VIPunit Clock Gating Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table>VIPunit Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>2</td><td><b>VINunit Clock Gating Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table>VINunit Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
</table>

UCGCTL3 - Unit Level Clock Gating Control 3

	1	**VFTunit Clock Gating Disable** Access: R/W VFTunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	0	**VFEunit Clock Gating Disable** Access: R/W VFEunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

UCGCTL4 - Unit Level Clock Gating Control 4

Register Space:	MMIO: 0/2/0
Default Value:	0x00F80003
Size (in bits):	32
Address:	0940Ch

Unit Level Clock Gating Control Registers.

DWord	Bit	Description
0	31:30	**Reserved** Access: RO Reserved.
	29	**GAFSRRB unit Clock Gate Disable** Access: R/W GAFSRRB units Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	28	**RAMDFT units Clock Gate Disable** Access: R/W RAMDFT units Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	27	**Reserved** Access: RO Reserved.
	26	**L3 CBR 1x Clock Gate Disable** Access: R/W L3 CBR units 1x Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	25	**L3 BANK 2x Clock Gate Disable** Access: R/W L3 BANK units 2x Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle

UCGCTL4 - Unit Level Clock Gating Control 4

		for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	24	**L3 BANK 1x Clock Gate Diable** Access: R/W L3 BANK units 1x Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	23	**MBGFunit Clock Gate Disable** Default Value: 1b Access: R/W MBGFunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	22	**MSQDunit 2x Clock Gate Disable** Default Value: 1b Access: R/W MSQD units cu2x Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	21	**MSQDunit Clock Gate Disable** Default Value: 1b Access: R/W MSQD units 1x Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	20	**MISDunits 2x Clock Gate Disable** Default Value: 1b Access: R/W MISDunits cu2x Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)

UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<tr><td></td><td></td><td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>19</td><td><b>MISDunit Clock Gate Disable</b><table><tr><td>Default Value:</td><td>1b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>MISDunits 1x Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>18</td><td><b>GAFMunit Clock Gate Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table>GAFMunit' Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>17</td><td><b>GAPCunit Clock Gate Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table>GAPCunits Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>16</td><td><b>GAPZunit Clock Gate Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table>GAPZunits' Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>15</td><td><b>GAPL3unit Clock Gate Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table>GAPL3 units' Clock Gating Disable Control:
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td></tr>
<tr><td></td><td>14</td><td><b>GAFSunit Clock Gate Disable</b><table><tr><td>Access:</td><td>R/W</td></tr></table></td></tr>
</table>

UCGCTL4 - Unit Level Clock Gating Control 4

		GAFSunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	13	**GAHSunit Clock Gate Disable** Access: R/W GAHSunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	12	**VISunit Clock Gate Disable** Access: R/W VISunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	11	**VACunit Clock Gate Disable** Access: R/W VACunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	10	**VAMunit Clock Gate Disable** Access: R/W VAMunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	9	**VADuit Clock Gating Disable** Access: R/W VADunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	8	**JPGunit Clock Gating Disable**

UCGCTL4 - Unit Level Clock Gating Control 4

		Access: R/W JPGunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	7	**VBPunits Clock Gating Disable** Access: R/W VBPunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	6	**VHRunit Clock Gating Disable** Access: R/W VHRunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	5	**VID4 VINunit Clock Gating Disable** Access: R/W VID4 VINunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	4	**VID3 VINunit Clock Gating Disable** Access: R/W VID3 VINunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	3	**VID2 VINunit Clock Gating Disable** Access: R/W VID2 VINunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

UCGCTL4 - Unit Level Clock Gating Control 4

	2	**VID1 VINunit Clock Gating Disable** Access: R/W VID1 VINunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
	1:0	**MSQCunit Clock Gating Disable** Default Value: 11b Access: R/W MSQCunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

CASCTLB_VLD_0 - Valid Bit Vector 0 for CASC

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04760h-04763h

This register contains the valid bits for entries 0-31 of CASCTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 0 for CASC** Default Value: 00000000h Access: RO Valid bits per entry

CVSTLB_VLD_0 - Valid Bit Vector 0 for CVS

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04724h-04727h

This register contains the valid bits for entries 0-31 of CVSTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 0 for CVS** Default Value: 00000000h Access: RO Valid bits per entry

L3TLB_VLD_0 - Valid Bit Vector 0 for L3

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04740h-04743h

This register contains the valid bits for entries 0-31 of L3TLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 0 for L3** Default Value: 00000000h Access: RO Valid bits per entry

MTTLB_VLD0 - Valid Bit Vector 0 for MTTLB

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04780h-04783h

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

VICTLB_VLD0 - Valid Bit Vector 0 for MTVICTLB

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04788h-0478Bh

This register contains the valid bits for entries 0-31 of VICTLB (Vertex Fetch, Instruction Cache, and Command Streamer TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

RCCTLB_VLD_0 - Valid Bit Vector 0 for RCC

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04728h-0472Bh

This register contains the valid bits for entries 0-31 of RCCTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 0 for RCC** Default Value: 00000000h Access: RO Valid bits per entry

RCCTLB_VLD0 - Valid Bit Vector 0 for RCCTLB

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04790h-04793h

This register contains the valid bits for entries 0-31 of RCCTLB (Render Cache for Color TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

RCZTLB_VLD0 - Valid Bit Vector 0 for RCZTLB

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04798h-0479Bh

This register contains the valid bits for entries 0-31 of RCZTLB (Render Cache for Z (Depth), Hi Z, and Stencil TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

MTTLB064_VLD0 - Valid Bit Vector 0 for TLB064

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	14780h-14783h

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

MTTLB132_VLD0 - Valid Bit Vector 0 for TLB132

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	14788h-1478Bh

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

MTTLB232_VLD0 - Valid Bit Vector 0 for TLB232

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	14790h-14793h

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

MTTLB304_VLD0 - Valid Bit Vector 0 for TLB304

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	14798h-1479Bh

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

TLBPEND_VLD0 - Valid Bit Vector 0 for TLBPEND registers

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04700h-04703h

This register contains the valid bits for entries 0-31 of TLBPEND structure (Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

VLFTLB_VLD_0 - Valid Bit Vector 0 for VLF

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04720h-04723h

This register contains the valid bits for entries 0-31 of VLFTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 0 for VLF** Default Value: 00000000h Access: RO Valid bits per entry

ZTLB_VLD_0 - Valid Bit Vector 0 for Z

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04730h-04733h

This register contains the valid bits for entries 0-31 of ZTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 0 for Z** Default Value: 00000000h Access: RO Valid bits per entry

CASCTLB_VLD_1 - Valid Bit Vector 1 for CASC

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04764h-04767h

This register contains the valid bits for entries 0-31 of CASCTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 1 for CASC** Default Value: 00000000h Access: RO Valid bits per entry

L3TLB_VLD_1 - Valid Bit Vector 1 for L3

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04744h-04747h

This register contains the valid bits for entries 0-31 of L3TLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 1 for L3** Default Value: 00000000h Access: RO Valid bits per entry

MTTLB_VLD1 - Valid Bit Vector 1 for MTTLB

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04784h-04787h

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLBVertex Fetch, Instruction Cache, and Command Streamer TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

MTVICTLB_VLD1 - Valid Bit Vector 1 for MTVICTLB

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	0478Ch-0478Fh

This register contains the valid bits for entries 0-31 of VICTLB (Vertex Fetch, Instruction Cache, and Command Streamer TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

RCCTLB_VLD_1 - Valid Bit Vector 1 for RCC

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	0472Ch-0472Fh

This register contains the valid bits for entries 0-31 of RCCTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 1 for RCC** Default Value: 00000000h Access: RO Valid bits per entry

RCCTLB_VLD1 - Valid Bit Vector 1 for RCCTLB

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04794h-04797h

This register is reserved for future RCC TLB extension.

DWord	Bit	Description
0	31:0	**Reserved** Format: MBZ

RCZTLB_VLD1 - Valid Bit Vector 1 for RCZTLB

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	0479Ch-0479Fh

This register contains the valid bits for entries 0-31 of RCZTLB (Render Cache for Z (Depth), Hi Z, and Stencil TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

MTTLB064_VLD1 - Valid Bit Vector 1 for TLB064

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	14784h-14787h

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

MTTLB132_VLD1 - Valid Bit Vector 1 for TLB132

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	1478Ch-1478Fh

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

MTTLB232_VLD1 - Valid Bit Vector 1 for TLB232

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	14794h-14797h

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

MTTLB304_VLD1 - Valid Bit Vector 1 for TLB304

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	1479Ch-1479Fh

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLB). Default Value = 00000000h Trusted Type = 1

DWord	Bit	Description
0	31:0	**Valid bits per entry**

TLBPEND_VLD1 - Valid Bit Vector 1 for TLBPEND registers

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	04704h-04707h

This register contains the valid bits for entries 32-63 of TLBPEND structure (Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

ZTLB_VLD_1 - Valid Bit Vector 1 for Z

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04734h-04737h

This register contains the valid bits for entries 0-31 of ZTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 1 for Z** Default Value: 00000000h Access: RO Valid bits per entry

CASCTLB_VLD_2 - Valid Bit Vector 2 for CASC

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04768h-0476Bh

This register contains the valid bits for entries 0-31 of CASCTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 2 for CASC** Default Value: 00000000h Access: RO Valid bits per entry

L3TLB_VLD_2 - Valid Bit Vector 2 for L3

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04748h-0474Bh

This register contains the valid bits for entries 0-31 of L3TLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 2 for L3** Default Value: 00000000h Access: RO Valid bits per entry

ZTLB_VLD_2 - Valid Bit Vector 2 for Z

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04738h-0473Bh

This register contains the valid bits for entries 0-31 of ZTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 2 for Z** Default Value: 00000000h Access: RO Valid bits per entry

CASCTLB_VLD_3 - Valid Bit Vector 3 for CASC

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	0476Ch-0476Fh

This register contains the valid bits for entries 0-31 of CASCTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 3 for CASC** Default Value: 00000000h Access: RO Valid bits per entry

L3TLB_VLD_3 - Valid Bit Vector 3 for L3

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	0474Ch-0474Fh

This register contains the valid bits for entries 0-31 of L3TLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 3 for L3** Default Value: 00000000h Access: RO Valid bits per entry

ZTLB_VLD_3 - Valid Bit Vector 3 for Z

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	0473Ch-0473Fh

This register contains the valid bits for entries 0-31 of ZTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 3 for Z** Default Value: 00000000h Access: RO Valid bits per entry

CASCTLB_VLD_4 - Valid Bit Vector 4 for CASC

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04770h-04773h

This register contains the valid bits for entries 0-31 of CASCTLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 4 for CASC** Default Value: 00000000h Access: RO Valid bits per entry

L3TLB_VLD_4 - Valid Bit Vector 4 for L3

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04750h-04753h

This register contains the valid bits for entries 0-31 of L3TLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 4 for L3** Default Value: 00000000h Access: RO Valid bits per entry

L3TLB_VLD_5 - Valid Bit Vector 5 for L3

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04754h-04757h

This register contains the valid bits for entries 0-31 of L3TLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 5 for L3** Default Value: 00000000h Access: RO Valid bits per entry

L3TLB_VLD_6 - Valid Bit Vector 6 for L3

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	04758h-0475Bh

This register contains the valid bits for entries 0-31 of L3TLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 6 for L3** Default Value: 00000000h Access: RO Valid bits per entry

L3TLB_VLD_7 - Valid Bit Vector 7 for L3

Register Space:	MMIO: 0/2/0
Default Value:	0x00000000
Size (in bits):	32
Address:	0475Ch-0475Fh

This register contains the valid bits for entries 0-31 of L3TLB

DWord	Bit	Description
0	31:0	**Valid Bit Vector 7 for L3** Default Value: 00000000h Access: RO Valid bits per entry

BCSTLB_VLD - Valid Bit Vector for BCS TLB

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24780h-24783h

This register contains the valid bits for entries 0-31 of BCS TLB.

DWord	Bit	Description
0	31:4	**Reserved**
	3:0	**Valid bits per entry**

BLBTLB_VLD - Valid Bit Vector for BLB TLB

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24784h-24787h

This register contains the valid bits for entries 0-31 of BLB TLB.

DWord	Bit	Description
0	31:8	**Reserved**
	7:0	**Valid bits per entry**

CTX_TLB_VLD - Valid Bit Vector for CTX TLB

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	24788h-2478Bh

This register contains the valid bits for entries 0-31 of CTX TLB.

DWord	Bit	Description
0	31:1	**Reserved**
	0	**Valid bits per entry**

PDTLB_VLD - Valid Bit Vector for PD TLB

Register Space:	MMIO: 0/2/0
Source:	BlitterCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	2478Ch-2478Fh

This register contains the valid bits for entries 0-31 of PD TLB.

DWord	Bit	Description
0	31:8	**Reserved**
	7:0	**Valid bits per entry**

VCS_CXT_SIZE - VCS Context Sizes

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00040D02
Access:	Read/32 bit Write Only
Size (in bits):	32
Address:	121A8h

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:21</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>20:16</td><td><b>VCS Context Size</b><table><tr><td>Format:</td><td>U5</td></tr></table><table><tr><th>Value</th><th>Name</th></tr><tr><td>4h</td><td><b>[Default]</b></td></tr></table></td></tr>
<tr><td></td><td>15:13</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>12:8</td><td><b>VCR Context Size</b><table><tr><td>Format:</td><td>U5</td></tr></table><table><tr><th>Value</th><th>Name</th></tr><tr><td>Dh</td><td><b>[Default]</b></td></tr></table></td></tr>
<tr><td></td><td>7:5</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>4:0</td><td><b>Reserved</b></td></tr>
</table>

VCS_CNTR - VCS Counter for the bit stream decode engine

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0xFFFFFFFF
Access:	R/W
Size (in bits):	32
Address:	12178h-1217Bh

DWord	Bit	Description
0	31:0	**Count Value** Default Value: ffffffffh Writing a Zero value to this register starts the counting. Writing a Value of FFFF FFFF to this counter stops the counter.

VCS_EIR - VCS Error Identity Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/WC
Size (in bits):	32
Address:	120B0h

The EIR register contains the persistent values of Hardware-Detected Error Condition bits. Any bit set in this register will cause the Master Error bit in the ISR to be set. The EIR register is also used by software to clear detected errors (by writing a 1 to the appropriate bit(s) except for the unrecoverable bits described).

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table>
</td></tr>
<tr><td></td><td>15:0</td><td><b>Error Identity Bits</b>
<table><tr><td>Format:</td><td>Array of Error condition bits ee the table titled Hardware-Detected Error Bits</td></tr></table>
This register contains the persistent values of ESR error status bits that are unmasked via the EMR register. The logical OR of all (defined) bits in this register is reported in the Master Error bit of the Interrupt Status Register. In order to clear an error condition, software must first clear the error by writing a 1 to the appropriate bit(s) in this field. If required, software should then proceed to clear the Master Error bit of the IIR.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td><b>[Default]</b></td><td></td></tr>
<tr><td>1h</td><td>Error occurred</td><td>Error occurred</td></tr>
</table>
<table>
<tr><th>Programming Notes</th></tr>
<tr><td>Writing a 1 to a set bit will cause that error condition to be cleared. However, the Page Table Error bit (Bit 4) cannot be cleared except by reset (i.e., it is a fatal error).</td></tr>
</table>
</td></tr>
</table>

VCS_EMR - VCS Error Mask Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x0000FFFF
Access:	R/W
Size (in bits):	32
Address:	120B4h

The EMR register is used by software to control which Error Status Register bits are "masked" or "unmasked". "Unmasked" bits will be reported in the EIR, thus setting the Master Error ISR bit and possibly triggering a CPU interrupt, and will persist in the EIR until cleared by software. "Masked" bits will not be reported in the EIR and therefore cannot generate Master Error conditions or CPU interrupts.

Undefined or reserved bits in the Hardware Detected Error Bit Table will always return a read value of '1'

DWord	Bit	Description
0	31:16	**Reserved** Default Value: 0000h Format: MBZ
	15:0	**Error Mask Bits** Format: Array of error condition mask bits See the table titled Hardware-Detected Error Bits. This register contains a bit mask that selects which error condition bits (from the ESR) are reported in the EIR.

Value	Name	Description
0000h	Not Masked	Will be reported in the EIR
FFFFh	Masked **[Default]**	Will not be reported in the EIR

VCS_ESR - VCS Error Status Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Address:	120B8h

The ESR register contains the current values of all Hardware-Detected Error condition bits (these are all by definition persistent). The EMR register selects which of these error conditions are reported in the persistent EIR (i.e., set bits must be cleared by software) and thereby causing a Master Error interrupt condition to be reported in the ISR.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>15:0</td><td><b>Error Status Bits</b><table><tr><td>Format:</td><td>Array of error condition bits See the table titled Hardware-Detected Error Bits.</td></tr></table>This register contains the non-persistent values of all hardware-detected error condition bits.<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td><b>[Default]</b></td><td></td></tr><tr><td>1h</td><td>Error Condition Detected</td><td>Error Condition detected</td></tr></table></td></tr>
</table>

VCS_EXCC - VCS Execute Condition Code Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W,RO
Size (in bits):	32
Trusted Type:	1
Address:	12028h

This register contains user defined and hardware generated conditions that are used by MI_WAIT_FOR_EVENT commands. An MI_WAIT_FOR_EVENT instruction excludes the executing ring from arbitration if the selected event evaluates to a 1, while instruction is discarded if the condition evaluates to a 0. Once excluded, a ring is enabled into arbitration when the selected condition evaluates to a 0.
This register also contains control for the invalidation of indirect state pointers on context restore.

DWord	Bit	Description
0	31:16	**Mask Bits** Format: Mask[15:0] These bits serve as a write enable for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.
	15:5	**Reserved** Format: MBZ
	4:0	**User Defined Condition Codes** The software may signal a Stream Semaphore by setting the Mask bit and Signal Bit together to match the bit field specified in a WAIT_FOR_EVENT (Semaphore).

VCS_HWSTAM - VCS Hardware Status Mask Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0xFFFFFFFF
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	12098h

Access: RO for Reserved Control bits

The HWSTAM register has the same format as the Interrupt Control Registers. The bits in this register are "mask" bits that prevent the corresponding bits in the Interrupt Status Register from generating a "Hardware Status Write" (PCI write cycle). Any unmasked interrupt bit (HWSTAM bit set to 0) will allow the Interrupt Status Register to be written to the ISR location (within the memory page specified by the Hardware Status Page Address Register) when that Interrupt Status Register bit changes state.

Programming Notes

- To write the interrupt to the HWSP, the corresponding IMR bit must also be clear (enabled).
- At most 1 bit can be unmasked at any given time.

DWord	Bit	Description
0	31:0	**Hardware Status Mask Register** Default Value: FFFFFFFFh Format: Array of Masks Refer to the table in the Interrupt Control Register section for bit definitions.

VCS_PWRCTX_MAXCNT - VCS IDLE Max Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000040
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	12054h

This register contains the time in 0.64us to wait before telling power management hardware the render pipe is IDLE

DWord	Bit	Description
0	31:20	**Reserved** Format: MBZ
	19:0	**MFX IDLE Wait Time** Format: Max Count Specifies how long the command stream should wait before ensuring the pipe is IDLE and to let power management hardware know

Value	Name	Description
00040h	**[Default]**	0x00040 * 0.64us ~ 41us wait time

Programming Notes

- This is only useable if bit 0 of the PC_PSMI_CTRL is clear.
- The value in this field *must* be greater than 1.

VCS_INSTPM - VCS Instruction Parser Mode Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	120C0h-120C3h

The VCS_INSTPM register is used to control the operation of the VCS Instruction Parser. Certain classes of instructions can be disabled (ignored) - often useful for detecting performance bottlenecks. Also, "Synchronizing Flush" operations can be initiated - useful for ensuring the completion (vs. only parsing) of rendering instructions. DefaultValue=0000 0000h

Programming Notes
All reserved bits are implemented.

DWord	Bit	Description
0	31:16	**Masks** Format: Mask[15:0] These bits serve as write enables for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.
	15:11	**Reserved** Format: MBZ
	10	**Reserved** Format: MBZ
	9	**TLB Invalidate** Format: U1 If set, this bit allows the command stream engine to invalidate the MFX TLBs. This bit is valid only with the Sync flush enable. **Note:** *GFX soft resets do not invalidate TLBs, it is up to GFX driver to explicitly invalidate TLBs post reset./*
	8:7	**Reserved** Format: MBZ
	6	**Memory Sync Enable** If set, this bit allows the video decode engine to write out the data from the local caches to memory.
	5	**Sync Flush Enable** Format: Enable (Cleared by HW)

VCS_INSTPM - VCS Instruction Parser Mode Register

<table>
<tr><td></td><td></td><td>This field is used to request a Sync Flush operation. The device will automatically clear this bit before completing the operation. See Sync Flush <i>(Programming Environment)</i>. Setting the Sync Flush Enable will cause a config write to MMIO register space with the address 0x4f100.
<table>
<tr><th>Programming Notes</th></tr>
<tr><td>The command parser must be stopped prior to issuing this command by setting the <b>Stop Ring</b> bit in register <b>BCS_MI_MODE</b>. Only after observing <b>Ring Idle</b> set in <b>BCS_MI_MODE</b> can a Sync Flush be issued by setting this bit. Once this bit becomes clear again, indicating flush complete, the command parser is re-enabled by clearing <b>Stop Ring</b>.</td></tr>
</table></td></tr>
<tr><td></td><td>4:0</td><td><b>Reserved</b>
<table>
<tr><td>Access:</td><td>R/W</td></tr>
<tr><td>Format:</td><td>MBZ</td></tr>
</table></td></tr>
</table>

VCS_IMR - VCS Interrupt Mask Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0xFFFFFFFF
Access:	R/W
Size (in bits):	32
Address:	120A8h

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

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:0</td><td><b>Interrupt Mask Bits</b>
<table><tr><td>Format:</td><td>Array of interrupt mask bits Refer to the Interrupt Control Register section for bit definitions.</td></tr></table>
This field contains a bit mask which selects which interrupt bits (from the ISR) are reported in the IIR.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>FFFF FFFFh</td><td><b>[Default]</b></td><td></td></tr>
<tr><td>0h</td><td>Not Masked</td><td>Will be reported in the IIR</td></tr>
<tr><td>1h</td><td>Masked</td><td>Will not be reported in the IIR</td></tr>
</table>
</td></tr>
</table>

VCS_MI_MODE - VCS Mode Register for Software Interface

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000200
Access:	R/W
Size (in bits):	32
Address:	1209Ch-1209Fh

The MI_MODE register contains information that controls software interface aspects of the command parser.

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31:16</td><td><b>Masks</b>
A 1 in a bit in this field allows the modification of the corresponding bit in Bits 15:0.</td></tr>
<tr><td></td><td>15</td><td><b>Suspend Flush</b>
<table><tr><td>Mask:</td><td>MMIO(0x209c)#31</td></tr></table>
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td>No Delay</td><td>HW will not delay flush, this bit will get cleared by MI_SUSPEND_FLUSH as well</td></tr>
<tr><td>1h</td><td>DelayFlush</td><td>Suspend flush is active</td></tr>
</table></td></tr>
<tr><td></td><td>14:12</td><td><b>Reserved</b>
<table><tr><td>Access:</td><td>R/W</td></tr></table></td></tr>
<tr><td></td><td>11</td><td><b>Invalidate UHPTR enable</b>
If bit set H/W clears the valid bit of BCS_UHPTR (4134h, bit 0) when current active head pointer is equal to UHPTR.</td></tr>
<tr><td></td><td>10</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>9</td><td><b>Ring Idle (Read Only Status bit)</b>
<table><tr><td>Access:</td><td>RO</td></tr></table>
<i>Writes to this bit are not allowed.</i>
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0</td><td>Parser not idle</td></tr>
<tr><td>1</td><td>Parser idle <b>[Default]</b></td></tr>
</table></td></tr>
<tr><td></td><td>8</td><td><b>Stop Ring</b>
Software must set this bit to force the Ring and Command Parser to Idle. Software must read a "1" in Ring Idle bit after setting this bit to ensure that the hardware is idle.

<i>Software must clear this bit for Ring to resume normal operation.</i>
<table>
<tr><th>Value</th><th>Name</th></tr>
<tr><td>0</td><td>Normal Operation</td></tr>
</table></td></tr>
</table>

VCS_MI_MODE - VCS Mode Register for Software Interface

		1	Parser is turned off
	7:0	**Reserved**	
		Access:	R/W

VCS_PP_DCLV - VCS PPGTT Directory Cacheline Valid Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Address:	12220h

This register controls update of the on-chip PPGTT Directory Cache during a context restore. Bits that are set will trigger the load of the corresponding 16 directory entry group.

This register is restored with context (prior to restoring the on-chip directory cache itself). This register is also restored when switching to a context whose LRCA matches the current CCID if the **Force PD Restore** bit is set in the context descriptor. The context image of this register must be updated and maintained by SW; SW should not normally need to read this register.

This register can also effectively be used to limit the size of a processes' virtual address space. Any access by a process that requires a PD entry in a set that is not enabled in this register will cause a fatal error, and no fetch of the PD entry will be attempted.

DWord	Bit	Description
0	63:32	**Reserved** Format: MBZ
	31:0	**PPGTT Directory Cache Restore [1..32] 16 entries** Format: Enable[32] If set, the [1st..32nd] 16 entries of the directory cache are considered valid and will be brought in on context restore. If clear, these entries are considered invalid and fetch of these entries will not be attempted.

VCS_TLBPEND_RDY0 - VCS Ready Bit Vector 0 for TLBPEND Registers

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	14708h-1470Bh

This register contains the ready bits for entries 0-31 of TLBPEND structure (Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Ready bits per entry**

VCS_TLBPEND_RDY1 - VCS Ready Bit Vector 1 for TLBPEND Registers

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	1470Ch-1470Fh

This register contains the ready bits for entries 32-63 of TLBPEND structure (Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Ready bits per entry**

VCS_TIMESTAMP - VCS Reported Timestamp Count

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000, 0x00000000
Access:	RO. This register is not set by the context restore.
Size (in bits):	64
Address:	12358h

This register provides an elapsed real-time value that can be used as a timestamp. This register is not reset by a graphics reset. It will maintain its value unless a full chipset reset is performed.

Note: This timestamp register reflects the value of the PCU TSC. The PCU TSC counts 10ns increments; this timestamp reflects bits 38:3 of the TSC (i.e. 80ns granularity, rolling over every 1.5 hours).

DWord	Bit	Description
0	63:36	**Reserved** Format: MBZ
	35:0	**Timestamp Value** Format: U36 This register toggles every 80 ns. The upper 28 bits are zero.

VCS_RNCID - VCS Ring Buffer Next Context ID Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Address:	12198h-1219Fh

This register contains the next ring context ID associated with the ring buffer.

Programming Notes
The current context (RCCID) register can be updated indirectly from this register on a context switch event. Note that this can only be triggered when arbitration is enabled or if the current context runs dry (head pointer becomes equal to tail pointer).

DWord	Bit	Description
0	63:0	**Context ID** See Context Descriptor for VCS.

VCS_TLBPEND_SEC0 - VCS Section 0 of TLBPEND Entry

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	14400h-14403h

This register is directly mapped to the TLBPEND Array in the Graphic Arbiter.

DWord	Bit	Description
0	31	**vtstatus** This bit will be used in conjunction with the ready bit to determine the stage of the translation. See table below.
	30:28	**GTT bits** Bits 3:1 of the GTT entry used to translate the Virtual Address. 000 if translation is pending.
	27:0	**Current address** The value of this field depends on the stage of the TLB translation for this entry: **VA** - bits 27:20 = 00, bits 19:0 = Bits 31:12 of the Virtual Address of the cycle.

VCS_TLBPEND_SEC1 - VCS Section 1 of TLBPEND Entry

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	14500h-14503h

This register is directly mapped to the current Virtual Addresses in the MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:28	**Current address** Bits 9:6 of the Virtual Address of the cycle.
	27:24	**Cacheability Control Bits** Bit 27 (bit 3 within the four-bit field) Reserved Bit 26 (bit 2 within the four-bit field) is the Graphics Data Type (GFDT) bit. It is the GFDT bit for this surface when writes occur. GFDT can also be set by the GTT. The effective GFDT is the logical OR of this field with the GFDT from the GTT entry. This field is ignored for reads. Bits 25:24 (bits 1:0 within the four-bit field) contain the Cacheability Control field, which controls cacheability as described in the following table: **Value** — **Description** 00b — Use cacheability control bits from GTT entry. 01b — Data is not cached. 11b — Data is cached.
	23	**ZLR bit** Flag to indicate this is a zero length read (A read used to calculate a Physical Address for a write).
	22:4	**TAG** Cycle identification TAG.

VCS_TLBPEND_SEC1 - VCS Section 1 of TLBPEND Entry

3:0 **SRC ID**

Encoding of unit generating this cycle

Constant	Value
SRCID	
VCS_RD_SRCID	"00000"
VMC_RD_SRCID	"00001"
VMX_RARD_SRCID	"00010"
VMX_BSRD_SRCID	"00011"
VMX_RSRD_SRCID	"00100"
VIP_RD_SRCID	"00101"
VLF_RD_SRCID	"00110"
VDS_ZLRD_SRCID	"00111"
VCS_WR_SRCID	"01000"
VMX_BSWR_SRCID	"01001"
VDS_WR_SRCID	"01010"
VOP_WR_SRCID	"01011"
VLF_RSWR_SRCID	"01100"
VLF_FDWR_SRCID	"01101"
VMX_RSWR_SRCID	"01110"
BSP_WR_SRCID	"01111"
VCR_RD_SRCID	"10001"
VCR_WR_SRCID	"10010"
VCS_RD_PROBE	"10011"

VCS_TLBPEND_SEC2 - VCS Section 2 of TLBPEND Entry

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	14600h-14603h

This register is directly mapped to the current Virtual Addresses in the MTTLB (Texture and constant cache TLB).

DWord	Bit	Description
0	31:11	**Reserved**
	10:8	**Current address** Bits 11:9 of the Virtual Address of the cycle.
	7:0	**PAT entry** Location of Physical Address in Physical Address Table.

VCS_THRSH - VCS Threshold for the counter of bit stream decode engine

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00150000
Access:	R/W
Size (in bits):	32
Address:	1217Ch-1217Fh

DWord	Bit	Description
0	31:0	**Threshold Value** Default Value: 00150000h The value in this register reflects the number of clocks the bit stream decode engine is expected to run. If the value is exceeded the counter is reset and an interrupt may be enabled in the device.

VCS_TLBPEND_VLD0 - VCS Valid Bit Vector 0 for TLBPEND Registers

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	14700h-14703h

This register contains the valid bits for entries 0-31 of TLBPEND structure (Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

VCS_TLBPEND_VLD1 - VCS Valid Bit Vector 1 for TLBPEND Registers

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	14704h-14707h

This register contains the valid bits for entries 32-63 of TLBPEND structure (Cycles pending TLB translation).

DWord	Bit	Description
0	31:0	**Valid bits per entry**

VFSKPD - VF Scratch Pad

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02470h
Address:	02740h-02743h

DWord	Bit	Description
0	31:16	**Mask Bits** Format: Mask[15:0] Must be set to modify corresponding bit in Bits 15:0. (All bits implemented)
	15	**Reserved** Format: MBZ
	14:7	**Reserved** Format: MBZ
	6	**Reserved** Format: MBZ
	5	**Reserved** Format: MBZ
	4:3	**Reserved** Format: MBZ
	2	**Vertex Cache Implicit Disable Inhibit** Format: U1 Value 0h, Name **[Default]**: Allow VF to disable VS0 when Sequential index or Prim ID is a valid Element. Value 1h: VF never implicitly disables the vertex cache. Software must disable the VS0 Cache when required.
	1	**Disable Over Fetch Cache** Format: MBZ This bit must be '0' always.
	0	**Reserved** Format: MBZ

Bit 2 values:

Value	Name	Description
0h	**[Default]**	Allow VF to disable VS0 when Sequential index or Prim ID is a valid Element.
1h		VF never implicitly disables the vertex cache. Software must disable the VS0 Cache when required.

VICTLB_VA - VIC Virtual page Address Registers

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000
Access:	RO
Size (in bits):	32
Trusted Type:	1
Address:	04900h-04903h

These registers are directly mapped to the current Virtual Addresses in the VICTLB (Vertex Fetch, Instruction Cache, and Command Streamer TLB.)

DWord	Bit	Description
0	31:12	**Address** Format: GraphicsAddress[31:12] Page virtual address.
	11:0	**Reserved** Format: MBZ

VBSYNC - Video/Blitter Semaphore Sync Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	12040h

This register is written by BCS, read by VCS.

DWord	Bit	Description
0	31:0	**Semaphore Data** Semaphore data for synchronization between video codec engine and blitter engine.

MFX_MODE - Video Mode Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS, VideoCS2
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	1229Ch

DWord	Bit	Description
0	31:16	**Mask Bits** Format: Mask[15:0] Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)
	15:14	**Reserved** Format: MBZ
	13:10	**Reserved** Format: MBZ
	9	**Per-Process GTT Enable** Format: Enable Per-Process GTT BS Mode Enable
	8	**Reserved**
	7	**Reserved** Format: MBZ
	6:5	**Reserved** Format: MBZ
	4:0	**Reserved** Format: MBZ

Bit 9 values:

Value	Name	Description
0h	PPGTT Disable **[Default]**	When clear, the Global GTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.
1h	PPGTT Enable	When set, the PPGTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.

VRSYNC - Video/Render Semaphore Sync Register

Register Space:	MMIO: 0/2/0
Source:	VideoCS
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Trusted Type:	1
Address:	12044h

This register is written by CS, read by VCS.

DWord	Bit	Description
0	31:0	**Semaphore Data** Semaphore data for synchronization between video codec engine and render engine.

VS_INVOCATION_COUNT - VS Invocation Counter

Register Space:	MMIO: 0/2/0
Source:	RenderCS
Default Value:	0x00000000, 0x00000000
Access:	R/W
Size (in bits):	64
Trusted Type:	1
Address:	02320h

This register stores the value of the vertex count shaded by VS. This register is part of the context save and restore.

DWord	Bit	Description
0	63:0	**VS Invocation Count Report** Number of vertices that are dispatched as threads by the VS stage. Updated only when **Statistics Enable** is set in VS_STATE (see the Vertex Shader Chapter in the 3D Volume.)

SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32

Address:	022D0h
Name:	RCS Wait For Event and Display Flip Flags Register
ShortName:	RCS_SYNC_FLIP_STATUS

This register is the saved value of what wait for events are still valid. This register is part of context save and restore for RC6 feature.

Programming Notes
Programming Restriction: This register should NEVER be programmed by SW, this is for HW internal use only.

DWord	Bit	Description
0	31	**Reserved** Format: MBZ
	30	**Display Plane A Asynchronous Display Flip Pending** Format: Enable This field enables a wait for the duration of a Display Plane A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	29	**Display Plane A Synchronous Flip Display Pending** Format: Enable This field enables a wait for the duration of a Display Plane A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	28	**Display Sprite A Synchronous Flip Display Pending** Format: Enable This field enables a wait for the duration of a Display Sprite A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
	27	**Reserved**

SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

		Format: MBZ
	26	**Display Plane B Asynchronous Display Flip Pending** Format: Enable This field enables a wait for the duration of a Display Plane B "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	25	**Display Plane B Synchronous Flip Display Pending** Format: Enable This field enables a wait for the duration of a Display Plane B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	24	**Display Sprite B Synchronous Flip Display Pending** Format: Enable This field enables a wait for the duration of a Display Sprite B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
	23	**Display Plane A Asynchronous Performance Flip Pending Wait Enable** Source: RenderCS Format: Enable This field enables a wait for the duration of a Display Plane A Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	23	**Reserved** Source: BlitterCS Format: MBZ
	22	**Display Plane A Asynchronous Flip Pending Wait Enable** Format: Enable This field enables a wait for the duration of a Display Plane A Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.

SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

<table>
<tr><td></td><td>21</td><td><b>Display Plane A Synchronous Flip Pending Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait for the duration of a Display Plane A Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>20</td><td><b>Display Sprite A Synchronous Flip Pending Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait for the duration of a Display Sprite A Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>19</td><td><b>Reserved</b><table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>18</td><td><b>Display Pipe A Scan Line Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait while a Display Pipe A Scan Line condition exists. This condition is defined as the the start of the scan line specified in the Pipe A Display Scan Line Count Range Compare Register. See Scan Line Event in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>17</td><td><b>Display Pipe A Vertical Blank Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait until the next Display Pipe A Vertical Blank event occurs. This event is defined as the start of the next Display Pipe A vertical blank period. Note that this can cause a wait for up to an entire refresh period. See Vertical Blank Event (See Programming Interface).</td></tr>
<tr><td></td><td>16</td><td><b>Display Pipe A H Blank Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait until the start of next Display Pipe A Horizontal Blank event occurs. This event is defined as the start of the next Display A Horizontal blank period. Note that this can cause a wait for up to a line. See Horizontal Blank Event in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>15</td><td><b>Display Plane B Asynchronous Performance Flip Pending Wait Enable</b><table><tr><td>Source:</td><td>RenderCS</td></tr><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait for the duration of a Display Plane B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front</td></tr>
</table>

SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

		buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	15	**Reserved** Source: BlitterCS Format: MBZ
	14	**Display Plane B Asynchronous Flip Pending Wait Enable** Format: Enable This field enables a wait for the duration of a Display Plane B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	13	**Display Plane B Synchronous Flip Pending Wait Enable** Format: Enable This field enables a wait for the duration of a Display Plane B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	12	**Display Sprite B Synchronous Flip Pending Wait Enable** Format: Enable This field enables a wait for the duration of a Display Sprite B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
	11	**Reserved** Format: MBZ
	10	**Display Pipe B Scan Line Wait Enable** Format: Enable This field enables a wait while a Display Pipe B Scan Line condition exists. This condition is defined as the the start of the scan line specified in the Pipe B Display Scan Line Count Range Compare Register. See Scan Line Event in the Device Programming Interface chapter of MI Functions.
	9	**Display Pipe B Vertical Blank Wait Enable** Format: Enable This field enables a wait until the next Display Pipe B Vertical Blank event occurs. This event is defined as the start of the next Display Pipe B vertical blank period. Note that this can cause a wait for up to an entire refresh period. See Vertical Blank Event (See Programming Interface).

SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

<table>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td>8</td><td><b>Display Pipe B H Blank Wait Enable</b>
<table><tr><td>Format:</td><td>Enable</td></tr></table>
This field enables a wait until the start of next Display Pipe B Horizontal Blank event occurs. This event is defined as the start of the next Display B Horizontal blank period. Note that this can cause a wait for up to a line. See Horizontal Blank Event in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td></td><td>7:5</td><td><b>Reserved</b>
<table><tr><td>Format:</td><td>MBZ</td></tr></table></td></tr>
<tr><td></td><td>4:0</td><td><b>Condition Code Wait Select</b>
This field enables a wait for the duration that the corresponding condition code is active. These enable select one of 15 condition codes in the EXCC register, that cause the parser to wait until that condition-code in the EXCC is cleared.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0h</td><td>Not Enabled</td><td>Condition Code Wait not enabled</td></tr>
<tr><td>1h-5h</td><td>Enabled</td><td>Condition Code select enabled; selects one of 5 codes, 0 - 4</td></tr>
<tr><td>6h-15h</td><td>Reserved</td><td></td></tr>
</table>
<table>
<tr><th>Programming Notes</th></tr>
<tr><td>Note that not all condition codes are implemented. The parser operation is UNDEFINED if an unimplemented condition code is selected by this field. The description of the EXCC register (Memory Interface Registers) lists the codes that are implemented.</td></tr>
</table></td></tr>
</table>

SYNC_FLIP_STATUS_1 - Wait For Event and Display Flip Flags Register 1

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32

Address:	022D4h
Name:	RCS Wait For Event and Display Flip Flags Register 1
ShortName:	RCS_SYNC_FLIP_STATUS_1

This register is the saved value of what wait for events are still valid. This register is part of context save and restore for RC6 feature.

DWord	Bit	Description
0	31:27	**Reserved** Format: MBZ
	26:15	**Reserved** Format: MBZ
	14:12	**Reserved** Format: MBZ
	11	**SyncFlush Status** Format: Enable This field toggles on completion of sync flush. This bit toggle generates Interrupt and also reports interrupt status to HWSP on sync flush done.
	10	**Display Plane C Asynchronous Display Flip Pending** Format: Enable This field enables a wait for the duration of a Display Plane C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	9	**Display Plane C Synchronous Flip Display Pending** Format: Enable This field enables a wait for the duration of a Display Plane C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	8	**Display Sprite C Synchronous Flip Display Pending** Format: Enable

SYNC_FLIP_STATUS_1 - Wait For Event and Display Flip Flags Register 1

		This field enables a wait for the duration of a Display Sprite C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
	7	**Display Plane C Asynchronous Performance Flip Pending Wait Enable** Source: RenderCS Format: Enable This field enables a wait for the duration of a Display Plane C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	7	**Reserved** Source: BlitterCS Format: MBZ
	6	**Display Plane C Asynchronous Flip Pending Wait Enable** Format: Enable This field enables a wait for the duration of a Display Plane C "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	5	**Display Plane C Synchronous Flip Pending Wait Enable** Format: Enable This field enables a wait for the duration of a Display Plane C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions.
	4	**Display Sprite C Synchronous Flip Pending Wait Enable** Format: Enable This field enables a wait for the duration of a Display Sprite C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
	3	**Reserved** Format: MBZ
	2	**Display Pipe C Scan Line Wait Enable**

SYNC_FLIP_STATUS_1 - Wait For Event and Display Flip Flags Register 1

<table>
<tr><td rowspan="3"></td><td></td><td><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait while a Display Pipe C Scan Line condition exists. This condition is defined as the the start of the scan line specified in the Pipe C Display Scan Line Count Range Compare Register. See Scan Line Event in the Device Programming Interface chapter of MI Functions.</td></tr>
<tr><td>1</td><td><b>Display Pipe C Vertical Blank Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait until the next Display Pipe C Vertical Blank event occurs. This event is defined as the start of the next Display Pipe C vertical blank period. Note that this can cause a wait for up to an entire refresh period. See Vertical Blank Event (See Programming Interface).</td></tr>
<tr><td>0</td><td><b>Display Pipe C H Blank Wait Enable</b><table><tr><td>Format:</td><td>Enable</td></tr></table>This field enables a wait until the start of next Display Pipe C Horizontal Blank event occurs. This event is defined as the start of the next Display C Horizontal blank period. Note that this can cause a wait for up to a line. See Horizontal Blank Event in the Device Programming Interface chapter of MI Functions.</td></tr>
</table>

PR_CTR_CTL - Watchdog Counter Control

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00000001
Access:	R/W
Size (in bits):	32
Address:	02178h
Name:	RCS Watchdog Counter Control
ShortName:	PR_CTR_CTL

<table>
<tr><th>DWord</th><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>31</td><td><b>Count Select</b>
Format: U1
<table><tr><th>Value</th><th>Name</th><th>Description</th></tr><tr><td>0h</td><td><b>[Default]</b></td><td>Use the timestamp to increment the watchdog count (every 640ns)</td></tr><tr><td>1h</td><td></td><td>Use the fixed function clock (csclk) to increment the watchdog count</td></tr></table></td></tr>
<tr><td></td><td>30:0</td><td><b>Counter Logic Op</b>
Default Value: 1h
This field specifies the action to be taken by the clock counter to generate interrupts.
Writing a Zero value to this register starts the counting.
Writing a Value of 0000_0001 to this counter stops the counter.</td></tr>
</table>

PR_CTR_THRSH - Watchdog Counter Threshold

Register Space:	MMIO: 0/2/0
Source:	BSpec
Default Value:	0x00145855
Access:	R/W
Size (in bits):	32

Address:	0217Ch
Name:	RCS Watchdog Counter Threshold
ShortName:	PR_CTR_THRSH

DWord	Bit	Description
0	31:0	**Counter Logic Threshold** Default Value: 00145855h Format: U32 This field specifies the threshold that the hardware checks against for the value of the render clock counter before generating an interrupt. The counter in hardware generates an interrupt when the threshold is reached, rolls over and starts counting again. The interrupt generated is the "Media Hang Notify" interrupt since this watchdog timer is intended primarily to remedy VLD hangs on the main pipeline.

WR_WATERMARK - Write Watermark

Register Space:	MMIO: 0/2/0
Default Value:	0x000FFEA4
Size (in bits):	32
Address:	04028h-0402Bh

Write Watermark

DWord	Bit	Description
0	31:20	**Counter Extra Bits** Default Value: 000000000000b Access: R/W Counter Extra Bits
	19	**Watermark Timeout Enable** Default Value: 1b Access: R/W Watermark timeout enable.
	18:8	**Watermark Timeout** Default Value: 11111111110b Access: R/W Number of clocks that the write pipe queue is allowed to keep a ready write cycle, without reads or writes to the queue. Once this value is met, and if the feature is enabled, the watermark will be considered reach, and all pending write requests will be issued.
	7	**Watermark En** Default Value: 1b Access: R/W Enable write request grouping
	6:0	**High Watermark**

WR_WATERMARK - Write Watermark

Default Value:	0100100b
Access:	R/W

This is the number of write requests to be collected before initiating a write burst. Once a burst is initiated, it will continue until all the available writes are requested.

ZTLB_LRA_0 - ZTLB LRA 0

Register Space:	MMIO: 0/2/0
Default Value:	0x1F107F00
Size (in bits):	32
Address:	04050h-04053h

ZTLB TLB LRA 0

DWord	Bit	Description
0	31	**Reserved** Default Value: 0b Access: RO Reserved Bits
	30:24	**ZTLB LRA1 Max** Default Value: 0011111b Access: R/W ZTLB LRA1 Max Format: U6 Maximum value of programmable LRA1
	23	**Reserved** Default Value: 0b Access: RO Reserved Format: U1
	22:16	**ZTLB LRA1 Min** Default Value: 0010000b Access: R/W ZTLB LRA1 Min Format: MBZ Minimum value of programmable LRA1
	15	**Reserved** Default Value: 0b Access: RO Reserved Bits

ZTLB_LRA_0 - ZTLB LRA 0

<table>
<tr><td></td><td>14:8</td><td><b>ZTLB LRA0 Max</b><table><tr><td>Default Value:</td><td>1111111b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>ZTLB LRA0 Max

Format: U1

Maximum value of programmable LRA0</td></tr>
<tr><td></td><td>7</td><td><b>Reserved</b><table><tr><td>Default Value:</td><td>0b</td></tr><tr><td>Access:</td><td>RO</td></tr></table>Reserved

Format: U1</td></tr>
<tr><td></td><td>6:0</td><td><b>ZTLB LRA0 Min</b><table><tr><td>Default Value:</td><td>0000000b</td></tr><tr><td>Access:</td><td>R/W</td></tr></table>ZTLB LRA0 Min

Format: U6

Minimum value of programmable LRA0</td></tr>
</table>

ZTLB_LRA_1 - ZTLB LRA 1

Register Space:	MMIO: 0/2/0
Default Value:	0x00002F20
Size (in bits):	32
Address:	04054h-04057h

ZTLB TLB LRA 1

DWord	Bit	Description
0	31:22	**Reserved** Default Value: 0000000000b Access: RO Reserved Format: MBZ
	21:20	**STC LRA** Default Value: 00b Access: R/W STC LRA Format: U6 Which LRA should STC use
	19:18	**HIZ LRA** Default Value: 00b Access: R/W HIZ LRA Format: U1 Which LRA should HIZ use
	17:16	**RCZ LRA** Default Value: 00b Access: R/W RCZ LRA Format: MBZ Which LRA should RCZ use
	15	**Reserved** Default Value: 0b

ZTLB_LRA_1 - ZTLB LRA 1

		Access: RO Reserved Bits
	14:8	**ZTLB LRA2 Max** Default Value: 0101111b Access: R/W ZTLB LRA2 Max Format: U1 Maximum value of programmable LRA2
	7	**Reserved** Default Value: 0b Access: RO Reserved Format: MBZ
	6:0	**ZTLB LRA2 Min** Default Value: 0100000b Access: R/W ZTLB LRA2 Min Format: U6 Minimum value of programmable LRA2