

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

Volume 2, Part 2: Command Reference - Instructions

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

© April 2014, Intel Corporation



Creative Commons License

You are free to Share — to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

No Derivative Works. You may not alter, transform, or build upon this work

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.

Copyright © 2014, Intel Corporation. All rights reserved.



Table of Contents

3DPRIMITIVE	9
3DSTATE_AA_LINE_PARAMETERS	
3DSTATE_BINDING_TABLE_POINTERS_DS	
3DSTATE_BINDING_TABLE_POINTERS_GS	
3DSTATE_BINDING_TABLE_POINTERS_HS	
3DSTATE_BINDING_TABLE_POINTERS_PS	
3DSTATE_BINDING_TABLE_POINTERS_VS	
3DSTATE_BLEND_STATE_POINTERS	
3DSTATE_CC_STATE_POINTERS	
3DSTATE_CHROMA_KEY	
3DSTATE_CLEAR_PARAMS	
3DSTATE_CLIP	
3DSTATE_CONSTANT_DS	
3DSTATE_CONSTANT_GS	
3DSTATE_CONSTANT_HS	
3DSTATE_CONSTANT_PS	
3DSTATE_CONSTANT_VS	
3DSTATE_DEPTH_BUFFER	
3DSTATE_DEPTH_STENCIL_STATE_POINTERS	
3DSTATE_DRAWING_RECTANGLE	
3DSTATE_DS	
3DSTATE_DS	
3DSTATE_GATHER_CONSTANT_HS	
3DSTATE_GS	
3DSTATE_GS	
3DSTATE_HIER_DEPTH_BUFFER	
3DSTATE_HS	
3DSTATE_HS	
3DSTATE_INDEX_BUFFER	
3DSTATE_LINE_STIPPLE	
3DSTATE_MONOFILTER_SIZE	
3DSTATE_MULTISAMPLE	
3DSTATE_POLY_STIPPLE_OFFSET	
3DSTATE_POLY_STIPPLE_PATTERN	
3DSTATE_PS	
3DSTATE_PUSH_CONSTANT_ALLOC_DS	
3DSTATE_PUSH_CONSTANT_ALLOC_GS	
3DSTATE_PUSH_CONSTANT_ALLOC_HS	
3DSTATE_PUSH_CONSTANT_ALLOC_PS	
3DSTATE_PUSH_CONSTANT_ALLOC_VS	

Command Reference - Instructions



3DSTATE_SAMPLE_MASK	
3DSTATE_SAMPLER_PALETTE_LOAD0	
3DSTATE_SAMPLER_PALETTE_LOAD1	
3DSTATE_SAMPLER_STATE_POINTERS_DS	
3DSTATE_SAMPLER_STATE_POINTERS_GS	
3DSTATE_SAMPLER_STATE_POINTERS_HS	
3DSTATE_SAMPLER_STATE_POINTERS_PS	
3DSTATE_SAMPLER_STATE_POINTERS_VS	
3DSTATE_SBE	
3DSTATE_SCISSOR_STATE_POINTERS	
3DSTATE_SF	
3DSTATE_SO_BUFFER	
3DSTATE_SO_DECL_LIST	
3DSTATE_STENCIL_BUFFER	
3DSTATE_STREAMOUT	
3DSTATE_TE	
3DSTATE_URB_DS	
3DSTATE_URB_GS	
3DSTATE_URB_HS	
3DSTATE_URB_VS	
3DSTATE_VERTEX_BUFFERS	
3DSTATE_VERTEX_ELEMENTS	
3DSTATE_VF_STATISTICS	
3DSTATE_VIEWPORT_STATE_POINTERS_CC	
3DSTATE_VIEWPORT_STATE_POINTERS_SF_CLIP	
3DSTATE_VS	
3DSTATE_WM	
add - Addition	
addc - Addition with Carry	
asr - Arithmetic Shift Right	
avg - Average	
bfe - Bit Field Extract	
bfi1 - Bit Field Insert 1	
bfi2 - Bit Field Insert 2	
bfrev - Bit Field Reverse	
brc - Branch Converging	
brd - Branch Diverging	
break - Break	
call - Call	
COLOR_BLT	
cmp - Compare	
cmpn - Compare NaN	
sendc - Conditional Send Message	



cont - Continue	
cbit - Count Bits Set	
dp2 - Dot Product 2	
dp3 - Dot Product 3	
dp4 - Dot Product 4	
dph - Dot Product Homogeneous	
else - Else	
endif - End If	
math - Extended Math Function	
fbl - Find First Bit from LSB Side	
fbh - Find First Bit from MSB Side	
frc - Fraction	
GPGPU_OBJECT	
GPGPU_WALKER	
f16to32 - Half Precision Float to Single Precision Float	
halt - Halt	
if - If	
illegal - Illegal	
subb - Integer Subtraction with Borrow	
jmpi - Jump Indexed	
Izd - Leading Zero Detection	
line - Line	
Irp - Linear Interpolation	
and - Logic And	
not - Logic Not	
or - Logic Or	
xor - Logic Xor	
MEDIA_CURBE_LOAD	
MEDIA_INTERFACE_DESCRIPTOR_LOAD	
MEDIA_OBJECT	
MEDIA_OBJECT_PRT	
MEDIA_OBJECT_WALKER	
MEDIA_STATE_FLUSH	
MEDIA_VFE_STATE	
MFC_AVC_PAK_OBJECT	
MFC_MPEG2_PAK_OBJECT	
MFC_MPEG2_SLICEGROUP_STATE	
MFD_AVC_BSD_OBJECT	
MFD_AVC_DPB_STATE	
MFD_AVC_SLICEADDR	
MFD_IT_OBJECT	
MFD_JPEG_BSD_OBJECT	

Command Reference - Instructions



MFD_MPEG2_BSD_OBJECT	
MFD_VC1_BSD_OBJECT	
MFD_VC1_LONG_PIC_STATE	
MFD_VC1_SHORT_PIC_STATE	
MFX_AVC_DIRECTMODE_STATE	
MFX_AVC_IMG_STATE	
MFX_AVC_REF_IDX_STATE	
MFX_AVC_SLICE_STATE	
MFX_AVC_WEIGHTOFFSET_STATE	
MFX_BSP_BUF_BASE_ADDR_STATE	
MFX_DBK_OBJECT	
MFX_FQM_STATE	
MFX_IND_OBJ_BASE_ADDR_STATE	
MFX_JPEG_HUFF_TABLE_STATE	
MFX_JPEG_PIC_STATE	
MFX_MPEG2_PIC_STATE	
MFX_PAK_INSERT_OBJECT	
MFX_PIPE_BUF_ADDR_STATE	
MFX_PIPE_MODE_SELECT	
MFX_QM_STATE	
MFX_STATE_POINTER	
MFX_STITCH_OBJECT	
MFX_SURFACE_STATE	
MFX_VC1_DIRECTMODE_STATE	
MFX_VC1_PRED_PIPE_STATE	
MFX_WAIT	
MI_ARB_CHECK	
MI_ARB_CHECK	
MI_ARB_CHECK	
MI_ARB_ON_OFF	
MI_ARB_ON_OFF	
MI_BATCH_BUFFER_END	
MI_BATCH_BUFFER_END	
MI_BATCH_BUFFER_END	
MI_BATCH_BUFFER_START	
MI_BATCH_BUFFER_START	
MI_BATCH_BUFFER_START	
MI_CLFLUSH	
MI_CONDITIONAL_BATCH_BUFFER_END	
MI_CONDITIONAL_BATCH_BUFFER_END	
MI_DISPLAY_FLIP	
MI_FLUSH	
MI FLUSH DW	



MI_FLUSH_DW	
MI_LOAD_REGISTER_IMM	
MI_LOAD_REGISTER_IMM	
MI_LOAD_REGISTER_IMM	
MI_LOAD_REGISTER_MEM	
MI_NOOP	
MI_NOOP	
MI_NOOP	
MI_PREDICATE	
MI_REPORT_HEAD	
MI_REPORT_HEAD	
MI_REPORT_HEAD	
MI_SEMAPHORE_MBOX	
MI_SEMAPHORE_MBOX	
MI_SEMAPHORE_MBOX	
MI_SET_CONTEXT	
MI_STORE_DATA_IMM	
MI_STORE_DATA_IMM	
MI_STORE_DATA_IMM	
MI_STORE_DATA_INDEX	
MI_STORE_DATA_INDEX	
MI_STORE_DATA_INDEX	
MI_STORE_REGISTER_MEM	
MI_TOPOLOGY_FILTER	
MI_URB_CLEAR	
MI_USER_INTERRUPT	
MI_USER_INTERRUPT	
MI_USER_INTERRUPT	
MI_WAIT_FOR_EVENT	
MI_WAIT_FOR_EVENT	
MI_WAIT_FOR_EVENT	
mov - Move	
movi - Move Indexed	
mul - Multiply	
mac - Multiply Accumulate	
mach - Multiply Accumulate High	
mad - Multiply Add	
nop - No Operation	
PIPE_CONTROL	
PIPELINE_SELECT	
pln - Plane	
ret - Return	

Command Reference - Instructions



rndd - Round Down	
rnde - Round to Nearest or Even	
rndz - Round to Zero	
rndu - Round Up	
sel - Select	
send - Send Message	
shl - Shift Left	
shr - Shift Right	
f32to16 - Single Precision Float to Half Precision Float	
SRC_COPY_BLT	
STATE_BASE_ADDRESS	
STATE_SIP	
sad2 - Sum of Absolute Difference 2	
sada2 - Sum of Absolute Difference Accumulate 2	
SWTESS_BASE_ADDRESS	
wait - Wait Notification	
while - While	
XY_COLOR_BLT	
XY_FULL_BLT	
XY_FULL_IMMEDIATE_PATTERN_BLT	
XY_FULL_MONO_PATTERN_BLT	
XY_FULL_MONO_PATTERN_MONO_SRC_BLT	
XY_FULL_MONO_SRC_BLT	
XY_FULL_MONO_SRC_IMMEDIATE_PATTERN_BLT	
XY_MONO_PAT_BLT	
XY_MONO_PAT_FIXED_BLT	
XY_MONO_SRC_COPY_BLT	
XY_MONO_SRC_COPY_IMMEDIATE_BLT	
XY_PAT_BLT_IMMEDIATE	
XY_PAT_BLT	
XY_PAT_CHROMA_BLT	
XY_PAT_CHROMA_BLT_IMMEDIATE	
XY_PIXEL_BLT	
XY_SCANLINES_BLT	
XY_SETUP_BLT	
XY_SETUP_CLIP_BLT	
XY_SETUP_MONO_PATTERN_SL_BLT	
XY_SRC_COPY_BLT	
XY_SRC_COPY_CHROMA_BLT	
XY_TEXT_BLT	
XY TEXT IMMEDIATE BLT	



3DPRIMITIVE						
Source:	Source: RenderCS					
Length E	Length Bias: 2					
The 3D Typicall The par function vertices	PRIMIT y the p amete n will u are th	TVE command is used to submorocessing results in rendering rs passed in this command are se this information to generate en passed down the 3D pipelir	it 3D primitives pixel data into tl forwarded to th e vertex data stru ne.	to be processed by the 3D pipeline. he render targets, but this is not rec e Vertex Fetch function. The Vertex uctures and store them in the URB.	iuired. Fetch These	
		Progra	mming Notes			
If the th threads that per Worklo	reads s spawn forms ads ena	pawned by this command are req ed from a previous command, sof a (preferably pipelined) memory fl abling topology filter using MI_TO	uired to observe tware must prece lush (e.g., 3D_PIPE POLOGY_FILTER n	memory writes performed by de this command with a command CONTROL). nust always program PIPECONTROL		
comma	nd with nd pro	only Post-Sync Operation (Write	Immediate data)	prior to every 3DPRIMTIVE		
DWord	Bit	granniea.	Descrip	tion		
0	31:29	Command Type				
-		Default Value:		3h GFXPIPE		
		Format: OpCode		OpCode		
	28:27	Command SubType				
		Default Value:	3h	GFXPIPE_3D		
		Format:	Op	Code		
	26:24	3D Command Opcode				
		Default Value: 3h 3DPRIMITIVE				
		Format:		OpCode		
	23:16	3D Command Sub Opcode				
		Default Value:	0h 3	DPRIMITIVE		
		Format:	OpC	ode		
	15:11	Reserved				
		Format:		MBZ		
	10	Indirect Parameter Enable		1		
		Format:		Enable		
		If set, the values in DW 2-5 are corresponding 3DPRIM_xxx M	MIO registers: (instead of DW2: \ nstead of DW3: St IT (instead of DW4 (instead of DW4 (instead of DW5: stead of DW6: Bas End Offset Enab	placed by the current values of the /ertexCountPerInstance) artVertexLocation) 4: InstanceCount) • StartInstanceLocation) eVertexLocation) ble must not be ENABLED at the sar	ne	



3DPRIMITIVE

	9	Reserved				
		Format:			MBZ	
	8	Predicate Enable			1	
		Format:		Enable	е	
		If set, this command is executed (or internal state bit. This command is ic bit is 0.	not) depending gnored only if Pr	on the edicat	e current value of the MI Predic teEnable is set and the Predicate	ate e state
	7:0	DWord Length				
		Default Value:	5h Excludes DV	Vord (0,1)	
		Format:	=n Total Lengt	h - 2		
1	31:10	Reserved			1	
		Format:			MBZ	
	9	End Offset Enable				
		Format:		Enable	e	
			Decemination			
	Description					
		register is used to indirectly spec	rify the vertex (COUNT	by defining the amount of	
		valid data in VB0. The following	restrictions app	ply:	by demining the amount of	
		VB0 must be enabled for use				
		VertexAccessType = SEQUEN	ITIAL			
		• Start Vertex Location = 0				
		• Start Instance Location = 0				
		• Base Vertex Location = 0				
		One added restriction applies:				
		• Instance Count = 1				
	Vertices are output until EndOffset is reached or exceeded in VB0. If EndOff is reached or exceeded within the data associated with a vertex, that vertex considered incomplete and will not be output. Partial objects will be discard (as is normally done). If clear, End Offset is ignored. Indirect Parameter Enable and End Offset Enable must not be ENABLED at same time, or behavior is UNDEFINED.					



				3DPRIMITIVE		
	8	Vertex This fiel Fetch.	Access Type d specifies ho	Type fies how data held in vertex buffers marked as VERTEXDATA is accessed by Vertex		
	Value Name Description				ription	
		0h	SEQUENTIAL	VERTEXDATA buffers are accessed so Enable is ENABLED.	equentiallyRequiref if End Offset	
		1h	RANDOM	VERTEXDATA buffers are accessed randomly via an index obtained from the Index Buffer.		
	7:6	Reserve	ed			
		Format	t:		MBZ	
	5:0	Primiti	ve Topology ⁻	Туре		
		Format	t: 3D_PrimTop general com	poType See table below for encoding, nments	see 3D Overview for diagrams and	
		This fiel single p triangle	d specifies the primitive topole s, etc.).	e topology type of 3D primitive generated by this command. Note that a logy (list/strip/fan/etc.) can contain a number of basic objects (lines,		
2	31:0	Vertex	Count Per Ins	stance		
		Format	t:	U32 Count of vertices		
		This field specifies how many vertices are to be generated for each instance of the primitive topology. If End Offset Enable is clear: Format = U32 count of vertices				
		Range = [0, 2^32-1] (upper limit probably constrained by VB size)				
	Ignored if End Offset Enable or Indirect Parameter Enable is ENABLED.				IS ENABLED.	
				Programming Note		
		•	This per-instant topology type vertices. Howe vertices will be	nce value should specify a valid numb e. E.g., for 3DPRIM_TRILIST_ADJ, this fi ever, in cases where too few or too m e silently discarded by the pipeline.	per of vertices for the primitive field should specify a multiple of 6 any vertices are provided, the unused	
		•	A 0 value is th	is field effectively makes the commar	nd a 'no-operation'.	



3DPRIMITIVE

3	31:0	Start Vertex Location				
		Format:	U32 structure index			
		This field specifies the "starting	ng vertex" for each instance. This al	llows skipping over part of the		
		vertices in a buffer if, for exar	nple, a previous 3DPRIMITIVE com	mand had already drawn the		
	primitives associated with the earlier entries.					
		For SEQUENTIAL access, this field specifies, for each instance, a starting structure index inter-				
		vertex butters	Id appaifing for each instance a sta	wing index into the Index Duffer		
		FOR RAINDOW access, this he	Programming Netes	arting index into the index buller.		
			Programming Notes			
		 Access of any data ou value 0 (i.e., appears a 	itside of the valid extent of a vertex as if the data stored at the invalid lo	c or index buffer will return the ocation was 0).		
		Must be set to 0 if En	d Offset Enable is ENABLED.			
		Ignored if Indirect Par	rameter Enable is ENABLED			
4	31:0	Instance Count				
		Format: U3	32 Count of instances			
			Description			
		This Gold as a Gold as the	Description	i sa da mada na inda da da s		
This field specifies the number of instances by which the primitive topology is to be						
		regenerated. A value of U indicates "no instances" (no-op operation). A value of 1				
		to provide instance data, if s	o programmed.			
		Ignored if Indirect Parameter Enable is ENABLED.				
		Must be set to 1 if End Offset Enable is ENABLED.				
			Value	Name		
		[0,FFFFFFFF]				
5	31:0	Start Instance Location				
		Format:	U32 structure index			
		Description				
		This field specifies the "starting instance" for the command as an initial structure index into INSTANCEDATA buffers.				
	tures, as controlled by					
		Programming Notes				
		Access of any data outside of the valid extent of a vertex or index buffer will return the				
		value 0 (i.e., appears a	as if the data stored at the invalid lo	ocation was 0).		
Must be set to 0 if End Offset Enable is ENABLED.						
Ignored if Indirect Parameter Enable is ENABLED.						
		Ignored if Indirect Pai	rameter Enable is ENABLED.			



6 31:0 Base Vertex Location Format: S31 index structure bias This field specifies a signed bias to be added to values read from the index buffer. This allows the same index buffer values to access different vertex data for different commands. This field applies only to RANDOM access mode. This field is ignored for SEQUENTIAL access mode, where there Start Vertex Location can be used to specify different regions in the vertex buffers. Programming Notes • Access of any data outside of the valid extent of a vertex or index buffer will return the value 0 (i.e., appears as if the data stored at the invalid location was 0). • Must be set to 0 if End Offset Enable is ENABLED. • Ignored if Indirect Parameter Enable is ENABLED.



3DSTA	re_aa_lini	E_PARAMETERS

 Source:
 RenderCS

 Length Bias:
 2

 The 3DSTATE_AA_LINE_PARAMS command is used to specify the slope and bias terms used in the improved alpha coverage computation (specifically for DX WHQL compliance). Note that in these devices the coverage values passed to PS threads are full U0.8 values.

values pa	assed to	o PS threads are full U0.8 values.										
DWord	Bit			Dese	riptio	on						
0	31:29	Command Type										
		Default Value:				3h GF	FXPIPE					
		Format:			OpCode							
	28:27	Command SubType										
		Default Value:			3h GF	XPIPE	_3D					
		Format:			ОрСо	ode						
	26:24	3D Command Opcode										
		Default Value: 1h 3DSTAT			NONP	PIPELIN	NED					
		Format:	0									
	23:16	D Command Sub Opcode										
		Default Value:	0Ah 3DSTATE_AA_LINE_PARAMS									
		Format:	OpCode									
	15:8	Reserved										
		Format:					MBZ					
	7:0	Dword Length		_								
		Default Value: 1h Ex			es Dw	ord (0	0,1)					
		Format:		=n Total I	ength	า - 2						
1	31:24	Reserved										
		Format: MBZ										
	23:16	AA Coverage Bias										
		Format:					U0.8					
		This field specifies the bias term to be used in the aa coverage computation for edges 0 and 3.										
	15:8	Reserved										
		Format:					MBZ					
	7:0	AA Coverage Slope										
		Format:					U0.8					
		This field specifies the slope te	rm to	be used in	the a	a cove	erage computation for edges 0 and 3.If					
		output expanded U0.8 coverad	will ie val	ues).	Jacy a	a ine (coverarge computation (though still					
		, , ,										
2	31:24	Reserved										
		Format:					MBZ					



	3DSTATE_AA_LINE_PARA	METERS							
23:16	AA Coverage EndCap Bias								
	Format:	U0.8							
	This field specifies the bias term to be used in the aa coverage computation for edges 1 and 2.								
15:8	Reserved								
	Format:	MBZ							
7:0	AA Coverage EndCap Slope								
	Format:	U0.8							
	This field specifies the slope term to be used in the aa coverage computation for edges 1 and 2.								



		3DSTATE_B	NDING	_TAB	SLE_	POINTERS_DS				
Source:		RenderCS								
Length B	ias:	2								
The 3DS BINDING	TATE_B 5_TABLE	INDING_TABLE_POINTEF E_STATE. Only some of th	RS_DS comma ne fixed functi	nd is used ions utilize	d to de e bindi	efine the location of fixed functions' ling tables.				
DWord	Bit			Des	criptio	on				
0	31:29	Command Type				-				
		Default Value:		3h GFXPIPE						
		Format:		OpCode						
	28:27	Command SubType	Command SubType							
		Default Value:	llue:			FXPIPE_3D				
		Format:		OpCo	ode					
	26:24	3D Command Opcode								
		Default Value:		0h 3DSTATE_PIPELINED						
		Format:		OpCode	OpCode					
	23:16	3D Command Sub Op	3D Command Sub Opcode							
		Default Value:	28h 3DSTAT	e_bindin	G_TAE	BLE_POINTERS_DS				
		Format:	OpCode							
	15:8	Reserved								
		Format: MBZ								
	7:0	DWord Length								
		Default Value:		0h DWORD_COUNT_n						
		Format:		=n						
1	31:16	Reserved								
		Format:				MBZ				
	15:5	Pointer to DS Binding	Table							
		Format: SurfaceS	tateOffset[15:	5]BINDIN	G_TAB	BLE_STATE*256				
		Specifies an aligned add relative to the Surface	dress offset of State Base A	f the funct ddress in	tion's l units c	BINDING_TABLE_STATE. This offset is of 32B.				
	4:0	Reserved								
		Format:				MBZ				



		3DSTATE_BI	NDING	_TAB	SLE_	PO	DINTERS_GS		
Source:		RenderCS							
Length B	ias:	2							
The 3DS BINDING	STATE_E S_TABLE	SINDING_TABLE_POINTE E_STATE. Only some of the	RS_GS comma ne fixed function	nd is use ons utilize	d to de e bindir	efine ng ta	the location of fixed functions' bles.		
DWord	Bit			Des	criptio	n			
0	31:29	Command Type							
		Default Value:				3h G	3h GFXPIPE		
		Format: Op			ОрС	ode			
	28:27	Command SubType							
		Default Value:		3h GF	XPIPI	E_3D			
		Format:		ОрСо	de				
	26:24	3D Command Opcode							
		Default Value:		0h 3DSTA	ATE_PIF	PELIN	IED		
		Format:		OpCode					
	23:16	3D Command Sub Ope	ode						
		Default Value:	29h 3DSTATE	E_BINDIN	G_TABI	LE_PO	DINTERS_GS		
		Format:	OpCode						
	15:8	Reserved							
		Format: MBZ							
	7:0	DWord Length							
		Default Value:		0h DWC	ORD_CO	OUN.	T_n		
		Format:		=n					
1	31:16	Reserved							
		Format:					MBZ		
	15:5	Pointer to GS Binding	Table						
		Format: SurfaceS	tateOffset[15:5	5]BINDIN	G_TABL	LE_ST	TATE*256		
		Specifies an aligned add	dress offset of	the funct	ion's B	INDI	NG_TABLE_STATE. This offset is		
			State Dase Au			I JZL			
	4:0	Reserved							
		Format:					MBZ		



		3DSTATE_BI	NDING	_TAB	LE_	POINTERS_HS			
Source:		RenderCS							
Length B	lias:	2							
The 3DS BINDING	STATE_E 5_TABLE	BINDING_TABLE_POINTE	RS_HS commane fixed functi	and is use ons utilize	d to d e bindi	define the location of fixed functions' ling tables.			
DWord	Bit			Des	criptic	on			
0	31:29	Command Type							
		Default Value:		3h GFXPIPE					
		Format:		OpCode					
	28:27	Command SubType							
		Default Value:	ault Value:			FXPIPE_3D			
		Format:		OpCc	ode				
	26:24	3D Command Opcode							
		Default Value:		0h 3DSTATE_PIPELINED					
		Format: OpCoc							
	23:16	3D Command Sub Op	code						
		Default Value:	27h 3DSTAT	e_bindin	G_TAB	BLE_POINTERS_HS			
		Format:	OpCode						
	15:8	Reserved							
		Format: MBZ							
	7:0	DWord Length							
		Default Value:		0h DWORD_COUNT_n					
		Format:		=n					
1	31:16	Reserved							
		Format:				MBZ			
	15:5	Pointer to HS Binding	Table						
		Format: SurfaceS	tateOffset[15:	5]BINDIN	G_TAB	BLE_STATE*256			
		Specifies an aligned add	dress offset of	f the funct	ion's l	BINDING_TABLE_STATE. This offset is			
			State Dase A	aress in	units c	UI 32D.			
	4:0	Reserved							
		Format:				MBZ			



		3DSTATE_BI	NDING		SLE_P	OINTERS_PS			
Source:		RenderCS							
Length B	lias:	2							
The 3DS BINDING	TATE_BI G_TABLE	NDING_TABLE_POINTER	S_PS commar le fixed function	nd is usec ons utilize	to define binding	e the location of fixed functions' tables.			
DWord	Bit			Des	ription				
0	31:29	Command Type							
		Default Value:		n GFXPIPE					
		Format: Op(pCode			
	28:27	ommand SubType							
		Default Value:		3h GFXP	IPE_3D				
		Format:							
	26:24	3D Command Opcode							
		Default Value:		0h 3DSTATE_PIPELINED					
		Format:		OpCode					
	23:16	3D Command Sub Opc	ode						
		Default Value:	2Ah 3DSTAT	e_bindin	G_TABLE	_POINTERS_PS			
		Format:	OpCode						
	15:8	Reserved							
		Format: MBZ							
	7:0	DWord Length							
		Default Value:		0h DWC	DRD_COU	JNT_n			
		Format:		=n					
1	31:16	Reserved							
		Format:				MBZ			
	15:5	Pointer to PS Binding	Table						
		Format: SurfaceSt	ateOffset[15:	5]BINDIN	G_TABLE_	_STATE*256			
		Specifies an aligned add	Iress offset of	the funct	ion's BIN	DING_TABLE_STATE. This offset is			
			olale Dase Au			2D.			
	4:0	Reserved							
		Format:				MBZ			



		3DSTATE_B	NDING	_TAE	SLE_	POINTERS_VS				
Source:		RenderCS								
Length B	ias:	2								
The 3DS BINDING	TATE_E	BINDING_TABLE_POINTE	RS_VS commane fixed function	and is use ions utilize	d to de e bindi	lefine the location of fixed functions' ling tables.				
DWord	Bit			Des	criptic	on				
0	31:29	Command Type								
		Default Value:		3h GFXPIPE						
		Format:		OpCode						
	28:27	Command SubType	Command SubType							
		Default Value:	efault Value:			FXPIPE_3D				
		Format:		OpCo	ode					
	26:24	3D Command Opcode								
		Default Value:		0h 3DSTATE_PIPELINED						
		Format:		OpCode						
	23:16	3D Command Sub Ope	code							
		Default Value:	26h 3DSTAT	e_bindin	G_TAB	BLE_POINTERS_VS				
		Format:	OpCode							
	15:8	Reserved								
		Format: MBZ								
	7:0	DWord Length								
		Default Value:		0h DWORD_COUNT_n						
		Format:		=n						
1	31:16	Reserved								
		Format:				MBZ				
	15:5	Pointer to VS Binding	Table							
		Format: SurfaceS	tateOffset[15:	5]BINDIN	G_TAB	3LE_STATE*256				
		Specifies an aligned add	dress offset of	f the funct	tion's E	BINDING_TABLE_STATE. This offset is				
		relative to the Surface :	State Base A	aaress in	units c	OT 32B.				
	4:0	Reserved								
		Format:				MBZ				



		3DST/	ATE_F	3LEN)_ST <i>F</i>	ATE	_ P (DINTERS		
Source:		RenderCS								
Length Bi	ias:	2								
The 3DS	TATE_B	LEND_STATE_POI	NTERS co	mmand is	used to se	et up tl	he po	inters to the color calculator sta	ate.	
			Pr	rogrammiı	ng Notes					
When th	e BLEN	D_STATE pointer	changes k	out not the	CC_STAT	E poin	ter, d	river needs to force a		
CC_STAT	CC_STATE pointer change to improve blend performance in pixel backend.									
DWord	Bit	ļ			Des	criptic	n			
0	31:29	Command Type								
		Default Value:					3h G	IFXPIPE		
		Format:	Format:				ОрСо	ode		
	28:27	Command SubT	уре			1				
		Default Value:			3h GF	XPIPE	E_3D			
		Format:				OpCode				
	26:24	3D Command O	pcode							
		Default Value:			Oh 3DST/	ATE_PI	PELIN	1ED		
		Format:			OpCode					
	23:16	3D Command S	ub Opcoc	de						
		Default Value:		24h 3DST	ATE_BLEN	ID_STA	ATE_PO	OINTERS		
		Format:		OpCode						
	15:8	Reserved						107		
		Format:				MBZ				
	7:0	DWord Length						Τ		
		Default value.				JRD_C	OUN	1_n		
1	21.6	Pland State Dei	- 4 a m		-11					
T	31:0	Blend State Poir	iter Dynamic	·StateOffse	+[31:6]BLF		ΓΔΤΕ*	¢Q		
		Specifies the 64	1-byte ali	igned offs	et of the	BLEN		o FATF This offset is relative to	the	
		Dynamic State	Base A	ddress		0.2.1.	0_0.		the	
	5:1	Reserved						[
		Format:						MBZ		
	0	Reserved						1		
		Format:						MB0		



		3D	STATE_	_CC_9	STAT	E_P	OINTERS			
Source:		RenderCS								
Length Bi	as:	2								
The 3DS	TATE_C	C_STATE_POINT	ERS comman	d is used	l to set up	the p	pointers to the color calculator st	ate.		
			Prog	grammiı	ng Notes					
When th	e CC_S ⁻	TATE pointer cha	anges but not	t the BLE	ND_STAT	E poin	ter, driver needs to force a			
BLEND_S	STATE p	ointer change ir	n order to imp	prove ble	end perfo	rmance	e in the pixel backend.			
DWord	Bit				Des	criptio	on			
0	31:29	Command Typ	e							
		Default Value:					3h GFXPIPE			
		Format:					OpCode			
	28:27	Command Sub	туре							
		Default Value:				3h GF	EXPIPE_3D			
		Format:			ОрСо	ode				
	26:24	3D Command	Opcode							
		Default Value:			0h 3DST/					
		Format:			OpCode					
	23:16	3D Command	Sub Opcode							
		Default Value:		0Eh 3DS	STATE_CC	_STAT	E_POINTERS			
		Format:		OpCode	5					
	15:8	Reserved								
		Format: MBZ								
	7:0	DWord Length	1		1					
		Default Value:			0h DWORD_COUNT_n					
		Format:			=n					
1	31:6	Color Calc Stat	te Pointer							
		Format:	DynamicStat	teOffset	31:6JCOL	DR_CA	ALC_STATE			
		Specifies the 6	Specifies the 64-byte aligned offset of the COLOR_CALC_STATE. This offset is relative							
			inc State Das	se Auui	633					
	5:1	Reserved								
		Format:					MBZ			
	0	Reserved								
		Format:					MBO			



		3DSTAT	E_CHRC)M/	4_k	(EY					
Source:		RenderCS									
Length E	Bias:	2									
The 3DS containin table ent Chromal supporte	TATE_C ng four try is as Key Ena ed.	CHROMA_KEY instruction is used t set of values is supported. The Cl ssociated with the map. Texture ch able texture sampler state variable	o program tex nromaKey Ind nromakey func . Texture Colc	cture co ex sam ctions a or Key (olor/o pler s are er keyin	chroma-key key values. A table state variable is used to select which nabled and controlled via use of the ng on a paletted texture index) is not					
DWord	Bit		Des	criptio	n						
0	31:29	Command Type									
		Default Value:		3h GFXPIPE							
		Format:			Орсо	ode					
	28:27	Command SubType	Command SubType								
		Default Value:		3h GFXPIPE_3D							
		Format:		Орсо	de						
	26:24	3D Command Opcode									
		Default Value:	1h 3DSTATE_	NONP	IPELII	NED					
		Format:									
2	23:16	3D Command Sub Opcode									
		Default Value:	04h 3DSTATE	_CHRC	DMA_	_KEY					
		Format:	Opcode								
	15:8	Reserved									
		Format:				MBZ					
	7:0	0 DWord Length									
		Default Value:	2h Exclud	des DWord (0,1)							
		Format:	=n								
		Total Length - 2									
1	31:30	ChromaKey Table Index									
		Format:		U2 ind	ex						
		Selects which entry in the Chrom	aKey table is t	o be lo	adec						
		Value				Name					
		[0,3]									
	29:0	Reserved				1					
		Format:				MBZ					
2	31:0	ChromaKey Low Value This field specifies the "low" (min considered "matching the key" if range. See ChromaKey High Valu	imum) value c each compon e for further f	of the c ent of ormat,	:hrom the to prog	na key range. Texel samples are exel falls within the (inclusive) chroma yramming info.					
3	31:0	ChromaKey High Value This field specifies the "high" (ma	ıximum) value	of the	chro	ma key range. Texel samples are					



	3DSTATE_CHROMA_KEY									
co rai	nsidered "matching the key' nge.	" if each	ר comp	onent	of th	e texel falls within the (inclusive) chroma				
	Programming Notes									
C th M	ChromaKey values are specified using 8-bit channels. When using surface formats with less than 8 bits per channel, the device will expand channels by replicating the required number of MSBs into the LSBs of each channel. Software must account for this conversion when it programs Chromakey Low/High Values (e.g., by performing the same replication).									
Fc sc ch C	For channels that do not exist in the actual surface (e.g., Alpha channel for non-ARGB maps), software must explicitly program full range high/low values (High=FFh, Low=Oh for formats using unsigned chroma key values, High=7Fh, Low=FFh for formats using sign magnitude chroma key values) in order to effectively remove the comparison of that field from the ChromaKey function.									
Fc ct pr R	or channels in SNORM format in the surface format, the value in the high/low value for that hannel is interpreted in sign magnitude format. Negative zero value is not supported (use positive zero instead). For channels with mixed UNORM/SNORM formats (i.e. 25G5 SNORM B6 UNORM), the ChromaKey is programmed as if all channels are SNORM.									
YI th ch	UV ChromaKey will use an ir ne chroma key values for tho hrominance value used is the	nterpola ose texe e avera	ated chi els witho ge of va	romina out ch alues t	ance v romir o the	value from the map for comparison to nance due to downsampling. The left and right of the texel in question.				
It	is UNDEFINED to program a prresponding component of	any con Chrom	nponen IaKey Lo	t of th ow Va	ne Chi lue.	romaKey High Value to be less than the				
Fo	ormat = interpreted accordi	ng to as	ssociate	ed texe	el forr	nat "class":				
Ous	only the surface formats listened with this feature. Use of	d as su any otł	pporteo ner surf	l for c ace fo	hrom rmat	a key in the surface formats table can be with chroma key enabled is UNDEFINED.				
	Surface Format	31:24	23:15	16:8	7:0					
A	ARGB and BC (DXT) formats	А	R	G	В					
	/CrCb formats	А	Cr	Y	Cb					



		3DSTATE	 C	LEAR	_PA	RAMS			
Source:		RenderCS							
Length B	ias:	2							
This com pipelinin	ımand o g isn't o	defines the depth clear value del completely transparent (see resti	ivereo rictior	d as a pipe n below).	lined	state command. However, the sta	te change		
	<u> </u>	Progra	ammi	na Notes					
Restriction: Prior to changing Depth/Stencil Buffer state (i.e., any combination of 3DSTATE_DEPTH_BUFFER, 3DSTATE_CLEAR_PARAMS, 3DSTATE_STENCIL_BUFFER, 3DSTATE_HIER_DEPTH_BUFFER) SW must first issue a pipelined depth stall (PIPE_CONTROL with Depth Stall bit set), followed by a pipelined depth cache flush (PIPE_CONTROL with Depth Flush Bit set, followed by another pipelined depth stall (PIPE_CONTROL with Depth Stall Bit set), unless SW can otherwise guarantee that the pipeline from WM onwards is already flushed (e.g., via a preceding MI_FLUSH). 3DSTATE_CLEAR_PARAMS must always be programmed in the along with the other Depth/Stencil state commands(i.e. 3DSTATE_DEPTH_BUFFER, 3DSTATE_STENCIL_BUFFER, or 3DSTATE_HIER_DEPTH_BUFFER)									
DWord	Bit			Desc	riptio	on			
0	31:29	Command Type				I .			
		Default Value:				3h GFXPIPE			
		Format:				OpCode			
	28:27	Command SubType							
		Default Value:			3h GF	EXPIPE_3D			
		Format:			ОрСо	ode			
	26:24	3D Command Opcode							
		Default Value:		0h 3DSTA	TE_PI	PELINED			
		Format:		OpCode					
	23:16	3D Command Sub Opcode							
		Default Value:	04h 3	BDSTATE_	CLEAR	PARAMS			
		Format:	ОрСо	ode					
	15:8	Reserved							
		Format:				MBZ			
	7:0	Dword Length							
		Default Value:		1h Exclud	es Dw	ord (0,1)			
		Format:		=n Total I	.ength	ו - 2			
1	31:0	Depth Clear Value							
		Format: for Surface Format of c IEEE_FloatD24_UNORM_S8_UIN UNORM in bits [23:0]D16_UNO	lepth IT: U2 IRM: l	buffer:D3 24 UNORM J16 UNOF	2_FLO I in bit M in I	AT_S8X24_UINT: IEEE_FloatD32_FL ts [23:0]D24_UNORM_X8_UINT: U2 bits [15:0]	OAT: 24		
		This field defines the clear value Clear field is enabled. It is valid	e that only	t will be ap if Depth B	plied uffer (to the depth buffer if the Depth B Clear Value Valid is set.	uffer		



3DSTATE_CLEAR_PARAMS

2	31:1	Reserved									
		Format:		MBZ							
	0	Depth Clear Value Valid	Depth Clear Value Valid								
		Format:	rmat: Boolean								
		This field enables the Depth Clear Value . If clear interpolated depth of an arbitrary pixel of the p WM_STATE or 3DSTATE_WM. If set, the depth of Value field of this command.	ar, the dept rimitive ren lear value i	th clear value is obtained from ndered with Depth Buffer Clear set in is obtained from the Depth Clear							



				3[DST	ATE	_CL	IP				
Source:		Rende	erCS									
Length E	Bias:	2										
DWord	Bit		Description									
0	31:29	29 Command Type										
		Default Valu	ue:					3h C	GFXPIPE			
		Format:						ОрС	Code			
	28:27	Command S	SubType									
		Default Valu	le:				3h (GFXPIP	E_3D			
		Format:					ОрО	Code				
	26:24	3D Comma	nd Opcode	1		1						
		Default Value: 0h 3						_PIPEL	INE			
		Format: OpCo					de					
	23:16	3D Comma	nd Sub Op	code		L						
		Default Valu	le:			12	2h 3DS	TATE_0	CLIP			
		Format: Op										
	15:8	Reserved							1			
		Format:							MBZ			
	7:0	DWord Len	gth									
		Default Value: 02h Exc					ides D	Word ((0,1)			
		Format:			=1	n Total	Lengtl	า - 2				
1	31:21	Reserved							1			
		Format:							MBZ			
	20	Front Winding Determines whether a triangle object is positions, when traversed in the order, r winding order. Does not apply to points Value Name Ob EBONITM			is considered "front facing" if the screen space vertex , result in a clockwise (CW) or counter-clockwise (CCW) its or lines. Description WINDING CW			e vertex ise (CCW)				
		1h			FRONT	WINDI	NG_C	CW				
	19	Vertex Sub	Pixel Preci	sion Se	elect							
		Format:								U1		
		Selects the r	number of f	raction	al bits n	naintair	ned in	the ver	rtex data			
		Value	Name				Des	criptic	on			
		0h		8 sub pixel pi			bits m	naintair	ned			
1h 4 sub pixel precision bits maintained												
	18	EarlyCull En	able									
		Format:						Enabl	e			
		This field is u	used to ena	ble/dis	able the	e EarlyC	ull fun	iction.				



				3	3DSTATE_CLIP					
	17:16	Cull Mo	de							
		Format			3D_CullMode					
		Controls triangle	removal (culli objects and do	ng) of Des no	of triangle objects based on orientation. The cull mode only a not apply to lines, points or rectangles.	pplies to				
		Value	Name		Description					
		0h	CULLMODE_B	ОТН	All triangles are discarded (i.e., no triangle objects are drawn)					
		1h CULLMODE_NONE No triangles are discarded due to orientation								
		2h	CULLMODE_F	RONT	T Triangles with a front-facing orientation are discarded					
		3h	CULLMODE_B	ACK	Triangles with a back-facing orientation are discarded					
					·					
					Programming Notes					
		Orienta	tion determina	ation i	is based on the setting of the Front Winding state.					
	15:11	Reserve	d							
		Format			MBZ					
	10	Clipper	Statistics Ena	ble						
		Format			Enable					
		This bit controls whether Clip-unit-specific statistics register(s) can be incremented.								
		Value	Name							
		0h	Disable	CL_IN	NVOCATIONS_COUNT cannot increment					
		lh	Enable	CL_IN	NVOCATIONS_COUNT can increment					
	9:8	Reserve	d							
		Format			IVIBZ					
	7:0	User Cli	p Distance Cu	II Tes	est Enable Bitmask					
		Format	t mark field co	locte	Enable[8]	/ trivial				
		accept c	etermination i	needs	ls to be made (does not cause a must clip). DX10 allows simu	Itaneous				
		use of C	lipDistance an	d Cull	Il Distance test of up to 8 distances.					
2	31	Clip Ena	ble							
		Format			Enable					
		Specifie	whether the	CLIP fu	function is enabled or disabled (pass-through).					
	30	API Mo	de							
		Controls	the definition	of the	he NEAR clipping plane					
		Value	Na	me	Description					
		0h	APIMODE_	OGL	NEAR VP boundary == 0.0 (NDC)					
		1h	Reserved							
	29	Reserve	d							
		Format			MBZ					



	Viewport XY ClipTest Enable								
	Forma	t:	Enable						
	This field is used to control whether the Viewport X,Y extents are considered in VertexClipTest.								
27	Viewpo	ort Z ClipTest Enable							
	Format: Enable								
	This fie Vertex(ld is used to control whet ClipTest.	her the Viewport Z extents (near, far) are considered in						
26	Guardband ClipTest Enable								
	Forma	t:	Enable						
	ENABLI both th with res	ED, ClipDetermination operation of the Guardband and Viewports spect to the XY directions	erates as if the Guardband were coincident with the Vie ort XY ClipTest are DISABLED, all vertices are considered						
25:24	Reserv	ed							
	Forma	t:	MBZ						
23:16	User Clip Distance Clip Test Enable Bitmask								
	Forma	t:	Enable[8]						
	This 8 k	pit mask field selects whic	h of the 8 user clip distances against which trivial reject						
15:13	This 8 t accept ClipDis Clip M	bit mask field selects whic / must clip determination tance and Cull Distance te ode Id specifies a general more	h of the 8 user clip distances against which trivial reject needs to be made. DX10 allows simultaneous use of est of up to 8 distances.						
15:13	This 8 t accept ClipDis Clip Ma This fie	bit mask field selects whic / must clip determination tance and Cull Distance te ode Id specifies a general mod	h of the 8 user clip distances against which trivial reject needs to be made. DX10 allows simultaneous use of est of up to 8 distances. de of the CLIP unit, when the CLIP unit is ENABLED.						
15:13	This 8 b accept ClipDis Clip Ma This fie Value Oh	bit mask field selects whic / must clip determination tance and Cull Distance te ode Id specifies a general mod Name CLIPMODE_NORMAL	h of the 8 user clip distances against which trivial reject needs to be made. DX10 allows simultaneous use of est of up to 8 distances. de of the CLIP unit, when the CLIP unit is ENABLED. Description TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded						
15:13	This 8 b accept ClipDis Clip M This fie Value Oh	bit mask field selects whic / must clip determination tance and Cull Distance te ode Id specifies a general mod Name CLIPMODE_NORMAL Reserved	h of the 8 user clip distances against which trivial reject needs to be made. DX10 allows simultaneous use of est of up to 8 distances. de of the CLIP unit, when the CLIP unit is ENABLED. Description TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded						
15:13	This 8 b accept ClipDis This fie Value Oh 1h 2h	bit mask field selects whic / must clip determination tance and Cull Distance te ode Id specifies a general mod Name CLIPMODE_NORMAL Reserved Reserved	h of the 8 user clip distances against which trivial reject needs to be made. DX10 allows simultaneous use of est of up to 8 distances. de of the CLIP unit, when the CLIP unit is ENABLED. Description TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded						
15:13	This 8 b accept ClipDis This fie Value Oh 1h 2h 3h	bit mask field selects whic / must clip determination tance and Cull Distance te ode Id specifies a general mod Name CLIPMODE_NORMAL Reserved Reserved CLIPMODE_REJECT_ALL	h of the 8 user clip distances against which trivial reject needs to be made. DX10 allows simultaneous use of est of up to 8 distances. de of the CLIP unit, when the CLIP unit is ENABLED. Description TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded All objects are discarded						
15:13	This 8 b accept ClipDis This fie Value Oh 1h 2h 3h 4h	bit mask field selects whic / must clip determination tance and Cull Distance te ode Id specifies a general mod Name CLIPMODE_NORMAL Reserved Reserved CLIPMODE_REJECT_ALL CLIPMODE_ACCEPT_ALL	h of the 8 user clip distances against which trivial reject needs to be made. DX10 allows simultaneous use of est of up to 8 distances.						



_	1-							
	Format:	MBZ						
9	Perspective Divide D	isable						
	Format: Disable							
	from the URB. This feature can be used by software to submit pre-transformed "screen-space" geometry for rasterization. This likely requires the W component of positions to contain "rhw" (aka 1/w) in order to support perspective-correct interpolation of vertex attributes. Likewise, th X,Y,Z components will likely be required to be X/W, Y/W, Z/W. Note that the device does not support clipping when perspective divide is disabled. Software must specify CLIPMODE_ACCEPT_ALL whenever it disables perspective divide. This implies that software must ensure that object positions are completely contained within the "guardband" screen-space limits imposed by the SE unit (a.g., by clipping in CPU SW before submitting the object).							
8	Non-Perspective Bar	vcentric Enable						
	Format:	Enable						
	are sent to SF unit in t barycentric parameter	he must clip case. This field must be enabled if any non-perspective s are enabled in the Windower.						
7:6	6 Reserved							
	Format:	MBZ						
5:4	Triangle Strip/List Provoking Vertex Select							
	Format:	U2 enumerated type						
	This field selects which "provoking vertex".	n vertex of a triangle (in a triangle strip or list primitive) is considere						
	Value	Name						
	0h	Vertex 0						
	1h	Vertex 1						
	2h	Vertex 2						
	3h	Reserved						
3:2	Line Strip/List Provo	king Vertex Select						
	Format:	U2 enumerated type						
	This field selects which vertex of a line (in a line strip or list primitive) is considered the "provoking vertex".							
	"provoking vertex".							
	"provoking vertex".	Name						
	"provoking vertex". Value	Name Vertex 0						
	"provoking vertex". Value Oh 1h	Name Vertex 0 Vertex 1						
	"provoking vertex". Value Oh 1h 2h	Name Vertex 0 Vertex 1 Reserved						
	This field selects which "provoking vertex". Uh Oh 1h 2h 3h	Name Vertex 0 Vertex 1 Reserved Reserved						
1:0	This field selects which "provoking vertex". Value Oh 1h 2h 3h Triangle Fan Provoki	Name Vertex 0 Vertex 1 Reserved Reserved Reserved						



			3DSTA	FE_CLIP		
		"provoking vertex".				
		Value		Name		
		0h	Vertex 0			
		1h	Vertex 1			
		2h	Vertex 2			
		3h	Reserved			
3	31:28	Reserved			1	
		Format:			MBZ	
	27:17	Minimum Point Width				
		Format:		U8.3 pixels		
		This value is used to clar	np read-back Poi	ntWidth values.		
	16:6	Maximum Point Width				
		Format:		U8.3 pixels		
		This value is used to clar	np read-back Poi	ntWidth values.		
	5	Force Zero RTAIndex E	nable			
		Format:		Enabl	е	
		If set, the Clip unit will ig	nore the read-ba	ack RTAIndex and	d opera	te as if the value 0 was read-
		back. If clear, the read-b	ack value is used.			
	4	Reserved				
		Format:			MBZ	
	3:0	Maximum VPIndex				
		Format: U4-	1 index value (#	of viewports)		
		This field specifies the m viewports. If the source passed down the pipelin URB.	aximum valid VP of the VPIndex ex ie. Note that this	Index value, corr ceeds this maxir clamping does r	esponc num va not affe	ing to the number of active lue, a VPIndex value of 0 is ct a VPIndex value stored in the



3DSTATE_CONSTANT_DS

Source: RenderCS

Length Bias:

This command sets pointers to the push constants for the DS unit. The constant data pointed to by this command is loaded into the DS unit's push constant buffer (PCB).

Programming Notes

It is invalid to execute this command more than once between 3D_PRIMITIVE commands.

Constant buffers must be enabled in order from Constant Buffer 0 to Constant Buffer 3 within this command. For example, It is not allowed to enable Constant Buffer 1 by programming a non-zero value in the DS Constant Buffer 1 Read Length without a non-zero value in DS Constant Buffer 0 Read Length.

DWord	Bit	Description							
0	31:29	Command Ty	уре						
		Default Value:					3h GFXPIPE		
		Format:					OpCode		
	28:27	Command Su	ubType						
		Default Value	e:			3h G	FXPIPE_3D		
		Format:				ОрСо	ode		
	26:24	3D Comman	d Opcode						
		Default Value	e:		0h 3DST	ATE_F	PIPELINED		
		Format:			OpCode				
	23:16	3D Comman	d Sub Opcode)					
		Default Value	e:	1Ah	1Ah 3DSTATE_CONSTANT_DS				
		Format:		Op0	OpCode				
	15:8	Reserved						1	
		Format:					MBZ		
	7:0	DWord Leng	th						
		Format:		=n To	n Total Length - 2				
								1	
		Value			Na	me		-	
		5n	Excludes DWG	ord (0,.	L) [Defau	Itj			
16	191:0	Constant Body							
		Format:	3DST/	ATE_CC	ONSTANT	(Body	()		
		Following tab and GS	le is the shared	d porti	on of the	3DST/	ATE_CONSTANT commar	ıd for VS, HS, DS,	



3DSTATE_CONSTANT_GS

Source: RenderCS

2

Length Bias:

This command sets pointers to the push constants for the GS unit. The constant data pointed to by this command will be loaded into the GS unit's push constant buffer (PCB).

Programming Notes

It is invalid to execute this command more than once between 3D_PRIMITIVE commands.

Constant buffers must be enabled in order from Constant Buffer 0 to Constant Buffer 3 within this command. For example, it is not allowed to enable Constant Buffer 1 by programming a non-zero value in the GS Constant Buffer 1 Read Length without a non-zero value in GS Constant Buffer 0 Read Length.

DWord	Bit			Des	criptio	on				
0	31:29	Command Ty	уре			-				
		Default Value	e:			3h GFXPIPE				
		Format:				OpCode				
	28:27	Command Su	Command SubType							
		Default Value	e:		3h G	FXPIPE_3D				
		Format:			ode					
	26:24	3D Comman	d Opcode	1			1			
		Default Value	e:	0h 3DS1	TATE_P	PIPELINED				
		Format:		OpCode	è					
	23:16	3D Comman	d Sub Opcode	1						
		Default Value	e:	16h 3DSTAT	E_CON	NSTANT_GS				
		Format:		OpCode						
	15	Reserved					1			
		Format:				MBZ				
	14:8	Reserved								
		Format:				MBZ				
	7:0	DWord Leng	th							
		Format:	=	n Total Lengtl	h - 2					
		Mahua		NL						
		Value Ch	Evoludos DW/or							
1.0	101.0		•	u (0,1) [Deiau	itj					
Т6	191:0	Constant Body								
		Following tab	SDSTAT	nortion of the) ATE CONSTANT comman	d for VS HS DS			
		and GS		portion of the	50517		u 101 v 3, 113, D3,			



3DSTATE_CONSTANT_HS

Source: RenderCS

Length Bias:

This command sets pointers to the push constants for the HS unit. The constant data pointed to by this command is loaded into the HS unit's push constant buffer (PCB).

Programming Notes

It is invalid to execute this command more than once between 3D_PRIMITIVE commands.

Constant buffers must be enabled in order from Constant Buffer 0 to Constant Buffer 3 within this command. For example, It is not allowed to enable Constant Buffer 1 by programming a non-zero value in the HS Constant Buffer 1 Read Length without a non-zero value in HS Constant Buffer 0 Read Length.

0 51.29 Command Type					
Default Value: 3h GFXPIPE					
Format: OpCode					
28:27 Command SubType					
Default Value: 3h GFXPIPE_3D	GFXPIPE_3D				
Format: OpCode					
26:24 3D Command Opcode					
Default Value: 0h 3DSTATE_PIPELINED					
Format: OpCode					
23:16 3D Command Sub Opcode					
Default Value: 19h 3DSTATE_CONSTANT_HS	.9h 3DSTATE_CONSTANT_HS				
Format: OpCode	pCode				
15:8 Reserved					
Format: MBZ	MBZ				
7:0 DWord Length					
Format: =n Total Length - 2					
Value Name					
Sh Excludes DWord (0,1) [Default]					
16 191:0 Constant Body]				
Format: 3DSTATE_CONSTANT(Body)					
Following table is the shared portion of the 3DSTATE_CONSTANT command for and GS	r VS, HS, DS,				



3DSTATE_CONSTANT_PS

Source: RenderCS

2

Length Bias:

This command sets pointers to the push constants for the PS unit. The constant data pointed to by this command is loaded into the PS unit's push constant buffer (PCB).

Programming Notes

It is invalid to execute this command more than once between 3D_PRIMITIVE commands.

Constant buffers must be enabled in order from Constant Buffer 0 to Constant Buffer 3 within this command. For example, it is not allowed to enable Constant Buffer 1 by programming a non-zero value in the PS Constant Buffer 1 Read Length without a non-zero value in PS Constant Buffer 0 Read Length.

DWord	Bit				Dese	criptio	on		
0	31:29	Command Type							
		Default Value:					3h GFXI	PIPE	
		Format:					OpCode	9	
	28:27	Command Su	ubType						
		Default Value	e:			3h Gl	XPIPE_3	D	
		Format:				ОрСо	ode		
	26:24	3D Comman	d Opcode						
		Default Value	e:		0h 3DST	ATE_P	IPELINE)	
		Format:			OpCode				
	23:16	3D Comman	d Sub Opcode	e					
		Default Value	e:	17	17h 3DSTATE_CONSTANT_PS				
		Format:		Ор	OpCode				
	15:8	Reserved							1
		Format:					ME	3Z	
	7:0	Dword Lengt	th	1					
		Format:		=n Tc	n Total Length - 2				
									1
		Value	Evelvele DW			me			
		51	Excludes DW	1) [Defau	tj				
16	191:0	Constant Body							
		Format:	3DST.	ATE_C	ONSTANT	(Body)		
		Following tab and GS	le is the share	d porti	ion of the	3DST/	ATE_CON	ISTANT comman	d for VS, HS, DS,



3DSTATE_CONSTANT_VS

Source: RenderCS

Length Bias: 2

This command sets pointers to the push constants for VS unit. The constant data pointed to by this command is loaded into the VS unit's push constant buffer (PCB).

Programming Notes

It is invalid to execute this command more than once between 3D_PRIMITIVE commands.

Constant buffers must be enabled in order from Constant Buffer 0 to Constant Buffer 3 within this command. For example, it is not allowed to enable Constant Buffer 1 by programming a non-zero value in the VS Constant Buffer 1 Read Length without a non-zero value in VS Constant Buffer 0 Read Length.

DWord	Bit		Description						
0	31:29	Command T	уре				_		
		Default Valu	e:				3h G	FXPIPE	
		Format:					OpCo	ode	
	28:27	Command S	ubType						
		Default Valu	e:			3h Gl	GFXPIPE_3D		
		Format:				ОрСо	ode		
	26:24	3D Comman	d Opcode						_
		Default Valu	e:		0h 3DST	ATE_P	PIPELI	NED	
		Format:			OpCode				
	23:16	3D Comman	d Sub Opcode						
		Default Valu	e:	15h	15h 3DSTATE_CONSTANT_VS				
		Format:		ОрС	OpCode				
	15:8	Reserved							
		Format:			MBZ				
	7:0	DWord Leng	jth						
		Format:	=	n Tot	al Length	ı - 2			
		Value		1 (0 1		me			
		5h	Excludes DWor	d (0,1) [Defaul	ltj			
16	191:0	Constant Body							
		Format:	3DSTAT	E_CO	NSTANT	(Body)		
		Following tab and GS	le is the shared _l	portic	on of the	3DST/	ATE_C	CONSTANT comman	d for VS, HS, DS,


			3DSTAT	E_[DEPTH	H_B	UF	FER		
Source:		Ren	derCS							
Length B	Bias:	2								
The depth buffer surface state is delivered as a pipelined state packet. However, the state change pipelining isn completely transparent (see restriction below).									ning isn't	
Programming Notes										
Restriction: Prior to changing Depth/Stencil Buffer state (i.e., any combination of 3DSTATE_DEPTH_BUFFER, 3DSTATE_CLEAR_PARAMS, 3DSTATE_STENCIL_BUFFER, 3DSTATE_HIER_DEPTH_BUFFER) SW must first issue a pipelined depth stall (PIPE_CONTROL with Depth Stall bit set), followed by a pipelined depth cache flush (PIPE_CONTROL with Depth Flush Bit set, followed by another pipelined depth stall (PIPE_CONTROL with Depth Stall Bit set), unless SW can otherwise guarantee that the pipeline from WM onwards is already flushed (e.g., via a preceding MI_FLUSH).										
3DSTAT commai 3DSTAT	E_DEPT nds(i.e. E_HIER	H_BUFFER 3DSTATE_ _DEPTH_B	t must always be progr CLEAR_PARAMS, 3DST UFFER).	amn ATE_	ned along _STENCIL_I	with th BUFFEF	ne otł R, or	her Depth/Stencil state	_	
The dep	oth buff	er is alway	vs Tile-Y							
DWord	Bit				Desc	riptio	n			
0	31:29	Command Type								
		Default V	alue:				3h G	JFXPIPE		
		Format:					ОрС	.ode		
	28:27	Comman	d SubType							
		Default V	alue:			3h GF	XPIPI	E_3D		
		Format:				OpCo	de			
	26:24	3D Comm								
		Default V	alue:							
		Format:			OpCode					
	23:16	3D Comm	nand Sub Opcode	0.51		DEDTI				
			alue:	05h	JSh JDSTATE_DEPTH_BUFFER					
		Format:		Ορι	Lode					
	15:8	Reserved						1407		
		Format:						MBZ		
	7:0	Dword Le	ength				1.77	0.1)		
		Default V	alue:		5h Exclud	es Dwo	ord ((0,1)		
Format: =n Total Length - 2										
1	31:29	Surface T This field	ype defines the type of the	surf	ace.					
		Value	Name					Description		
		0h	SURFTYPE_1D	De	efines a 1-c	dimens	iona	l map or array of maps		
1h SURFTYPE_2D Defines a 2-dimensional map or array of						l map or array of maps				



	-									
	2h	SURFTYPE_3D	Defines a 3-dimens	Defines a 3-dimensional (volumetric) map						
	3h	SURFTYPE_CUBE	Defines a cube map)						
	4h-6h	Reserved								
	7h	ce								
			Programming	Notes						
	The Surfa	Programming Notes								
	target(s) (defined in SURFACE_STATE), unless either the depth buffer or render targets are SURFTYPE_NULL									
28	Depth Wr	Depth Write Enable								
	Format:		Er	nable						
	This field e Write Ena	This field enables depth writes to the depth buffer surface. Both this field and the Depth Buffer Write Enable field in DEPTH_STENCIL_STATE must be enabled in order for depth writes to occur.								
27	Stencil W	rite Enable								
	Format:		Er	nable						
	This field e stencil is lo DEPTH_ST	enables stencil writes t ocated. Both this field ENCIL_STATE must be	o the depth buffer or and the Stencil Buffe enabled in order for s	stencil buffer surface, depending on whe r Write Enable field in stencil writes to occur.	re					
26:23	Reserved									
	Format:		MBZ							
22	Hierarchio	al Depth Buffer Ena	ble							
	Format:		Enable							
	If enabled,	indicates that a hiera	rchical depth buffer is	licates that a hierarchical depth buffer is defined.						
		Programming Notes								
			Programming I	Notes						
	If this field be disable	d is enabled, the Soft ed if Early Depth Test	Programming I ware Tiled Rendering Enable is disabled.	Notes J Mode must be NORMAL. This field mus	t					
21	If this field be disable Reserved	d is enabled, the Soft e ed if Early Depth Test	Programming I ware Tiled Rendering : Enable is disabled.	Notes 9 Mode must be NORMAL. This field mus	t					
21	If this field be disable Reserved Format:	d is enabled, the Soft ed if Early Depth Test	Programming I ware Tiled Rendering : Enable is disabled.	Notes 9 Mode must be NORMAL. This field mus	t					
21 20:18	If this field be disable Reserved Format: Surface Fo Specifies the field for reserved	d is enabled, the Soft ed if Early Depth Test Depth Test he format strictions on the dept strictions on the use o	Programming I ware Tiled Rendering : Enable is disabled. h buffer. See Stencil T of some of these formation	Notes g Mode must be NORMAL. This field mus MBZ Test Enable field in DEPTH_STENCIL_STAT ats.	it 					
21 20:18	If this field be disable Reserved Format: Surface Fo Specifies th field for re Value	d is enabled, the Soft ed if Early Depth Test ormat he format of the dept strictions on the use o	Programming I ware Tiled Rendering Enable is disabled. The buffer. See Stencil T of some of these formation lame	Notes g Mode must be NORMAL. This field mus MBZ Test Enable field in DEPTH_STENCIL_STAT ats. Description	it ΓΕ					
21 20:18	If this field be disable Reserved Format: Surface Fo Specifies th field for re Value Oh	d is enabled, the Soft ed if Early Depth Test ormat he format of the dept strictions on the use of Reserved	Programming I ware Tiled Rendering t Enable is disabled. h buffer. See Stencil T of some of these forma lame	Notes g Mode must be NORMAL. This field must MBZ Gest Enable field in DEPTH_STENCIL_STAT ats. Description Reserved	E					
21 20:18	If this field be disable Format: Surface Fo Specifies the field for re Value Oh 1h	d is enabled, the Soft ed if Early Depth Test ormat he format of the depti strictions on the use of Reserved D32_FLOAT	Programming I ware Tiled Rendering t Enable is disabled. h buffer. See Stencil T of some of these forma lame	Notes g Mode must be NORMAL. This field must MBZ Gest Enable field in DEPTH_STENCIL_STAT ats. Description Reserved D32_FLOAT	E					
21 20:18	If this field be disable Reserved Format: Surface Fo Specifies the field for ree Value Oh 1h 2h	d is enabled, the Soft ed if Early Depth Test ormat he format of the dept strictions on the use of Reserved D32_FLOAT Reserved	Programming I ware Tiled Rendering t Enable is disabled. h buffer. See Stencil T of some of these forma lame	Notes g Mode must be NORMAL. This field must MBZ Test Enable field in DEPTH_STENCIL_STAT ats. Description Reserved D32_FLOAT Reserved	E					
21 20:18	If this field be disable Reserved Format: Surface Fo Specifies the field for re Value Oh 1h 2h 3h	d is enabled, the Soft ed if Early Depth Test ormat he format of the dept strictions on the use of Reserved D32_FLOAT Reserved D24_UNORM_X8_U	Programming I ware Tiled Rendering t Enable is disabled. h buffer. See Stencil T of some of these forma lame	Notes g Mode must be NORMAL. This field must MBZ Gest Enable field in DEPTH_STENCIL_STAT ats. Description Reserved D32_FLOAT Reserved D24_UNORM_X8_UINT	E					
21 20:18	If this field be disable Reserved Format: Surface Fo Specifies the field for re Value Oh 1h 2h 3h 4h	d is enabled, the Soft ed if Early Depth Test be format of the depti strictions on the use of Reserved D32_FLOAT Reserved D24_UNORM_X8_U Reserved	Programming I ware Tiled Rendering t Enable is disabled. h buffer. See Stencil T of some of these forma lame	Notes g Mode must be NORMAL. This field must MBZ Test Enable field in DEPTH_STENCIL_STAT ats. Description Reserved D32_FLOAT Reserved D24_UNORM_X8_UINT Reserved	E					
21 20:18	If this field be disable Reserved Format: Surface Fo Specifies the field for ree Value Oh 1h 2h 3h 4h 5h	d is enabled, the Soft ed if Early Depth Test be format of the dept strictions on the use of D32_FLOAT Reserved D24_UNORM_X8_U Reserved D16_UNORM	Programming I ware Tiled Rendering t Enable is disabled. h buffer. See Stencil T of some of these forma lame	Notes g Mode must be NORMAL. This field must MBZ Test Enable field in DEPTH_STENCIL_STAT ats. Description Reserved D32_FLOAT Reserved D24_UNORM_X8_UINT Reserved D16_UNORM	E					



			3DS	TATE_DEPTH_BUFFER						
	17:0	Surface Pitch								
		Format: U18-1 Pitch in Bytes								
		This field specifie	es the p	itch of the depth buffer in (#Bytes - 1).						
		Value	Name	Description						
		[127, 3FFFFh]		corresponding to [128B, 256KB] also restricted to a multiple of 128B						
				Programming Notes						
		The pitch specified must be a multiple of the tile pitch, in the range [128B, 128KB].								
2	31:0	Surface Base Ac	dress							
		Format:	Gra	ohicsAddress[31:0]Depth_Buffer						
		This field specifie	es the s	tarting Dword address of the buffer in mapped Graphics Memory.						
				Programming Notes						
		The Depth Buffe	er can o	nly be mapped to Main Memory (uncached).						
		If the buffer is li	near, th	e surface must be 64-byte aligned.						
3	31:18	Height								
		Format:		U14						
		Range: SURFTY	PE_1D: I	nust be zeroSURFTYPE_2D: height of surface - 1 (y/v dimension)						
		[0,16383]SURFT of surface - 1 (y	YPE_3D /v dime	: height of surface - 1 (y/v dimension) [0,2047]SURFIYPE_CUBE: height ension) [0, 16383]						
		This field specifi height of the ba	es the se MIP	neight of the surface. If the surface is MIP-mapped, this field contains the level.						
				Programming Notes						
		The Height of th in SURFACE_STA (non-array) and	ne dept ATE), ur LOD =	h buffer must be the same as the Height of the render target(s) (defined less Surface Type is SURFTYPE_1D or SURFTYPE_2D with Depth = 0 0 (non-mip mapped).						
	17:4	Width								
		Format:		U14-1						
		Range: SURFTYPE_1D: width of surface - 1 (x/u dimension) [0, 16383]SURFTYPE_2D: width of surface - 1 (x/u dimension) [0, 16383]SURFTYPE_3D: width of surface - 1 (x/u dimension) [0, 2047]SURFTYPE_CUBE: width of surface - 1 (x/u dimension) [0, 16383]								
		This field specifi width of the bas	es the se MIP	width of the surface. If the surface is MIP-mapped, this field specifies the evel. The width is specified in units of pixels.						
				Programming Notes						
		The Width spec bytes via the Su of the depth bu SURFACE_STATI array) and LOD	ified by rface Pi ffer mu E), unles = 0 (no	this field must be less than or equal to the surface pitch (specified in tch field). For cube maps, Width must be set equal to Height.The Width st be the same as the Width of the render target(s) (defined in ss Surface Type is SURFTYPE_1D or SURFTYPE_2D with Depth = 0 (non-n-mip mapped).						



3DSTATE_DEPTH_BUFFER

	3:0	LOD							
		Format:		U4 in LOD uni	ts				
		This field define	s the MIP level that	at is currently b	eing rend	ered into.			
			Value			Name			
		[0, 14]	[0, 14]						
		Programming Notes							
		The LOD of the depth buffer must be the same as the LOD of the render target(s) (defined in SURFACE_STATE)							
4	31:21	 Depth							
		Format:	Format: U11-1						
		This field specifies the total number of levels for a volume texture or the number of array elements allowed to be accessed starting at the Minimum Array Element for arrayed surfaces. If the volume texture is MIP-mapped, this field specifies the depth of the base MIP level.							
		Value			Nam	e			
		[0, 2047]	SURFTYPE_1D nu	mber of array e	elements -	1			
		[0, 2047]	SURFTYPE_2D nu	mber of array e	elements -	1			
		[0, 2047]	SURFTYPE_3D de	pth of surface	- 1 (r/z din	nension)			
		0 SURFTYPE_CUBE (must be zero)							
		Programming Notes							
		The Denth of t	The Depth of the depth buffer must be the same as the Depth of the render target(s) (defined						
		in SURFACE_STATE).							
	20:10	Minimum Array Element							
		Format:				U11			
		For 1D and 2D Surfaces: This field indicates the minimum array element that can be accessed as part of this surface. The delivered array index is added to this field before being used to address the surface. For 3D Surfaces: This field indicates the minimum 'R' coordinate on the LOD currently being rendered to. This field is added to the delivered array index before it is used to address the surface. For Other Surfaces: This field is ignored							
		V	alue			Name			
		[0, 2047]		SURFTYPE_1D	/2D				
		[0, 2047]		SURFTYPE_3D					
	9:4	Reserved							
		Format:				MBZ			
	3:0	Depth Buffer C		ate	STATE				
		Specifies the m	IVIEIVIORI_OBJ	rol state for the	A	uffer			
		specifies the me	Specifies the memory object control state for the depth buffer.						



		3DSTATE_DEPTH_BUFFER						
5	31:16	Depth Coordinate Offset Y						
		Format: S15 in Screen Space (pixels)(3 LSBs MBZ)						
		Specifies a signed pixel offset to be added to the RenderTarget Y coordinate in order to						
		generate a Deptilbuner i coordinate. (See Deptil coordinate in Windower).						
		Programming Notes						
		The 3 LSBs of both offsets must be zero to ensure correct alignmentSoftware must ensure that						
		This field must be zero when rendering to field-mode (interlaced) Color Buffers (i.e., when						
		Surface State's VerticalLineStride==1).						
		This field can only be nonzero when rendering to surfaces of type SURFTYPE_1D and SURFTYPE_2D with Depth = 0 (non-array) and LOD = 0 (non-mip mapped).						
		The offsets need to be aligned to the hashing mode set for WM in the GT_MODE register (0x7008) bits[12:10].						
		For eg if the hashing mode is set to 16x16, the Depth Coordinate Y offset needs to be aligned to the 16x16 pixel block.						
	15:0	Depth Coordinate Offset X						
		Format: S15 in Screen Space (pixels)(3 LSBs MBZ)						
		Range: [-8192,8191] Bits 15:14 should be a sign extension						
		Specifies a signed pixel offset to be added to the RenderTarget X coordinate in order to generate a DepthBuffer X coordinate. (See Depth Coordinate in Windower).						
		Programming Notes						
		The 3 LSBs of both offsets must be zero to ensure correct alignmentSoftware must ensure that						
		the resulting (sum) coordinate value is non-negative.						
		This field must be zero when rendering to field-mode (interlaced) Color Buffers (i.e., when						
		This field can only be nonzero when rendering to surfaces of type SURFTYPE 1D and						
		SURFTYPE_2D with Depth = 0 (non-array) and LOD = 0 (non-mip mapped).						
		The offsets need to be aligned to the hashing mode set for WM in the GT_MODE register						
		(0x/008) bits[12:10]. For eq if the bashing mode is set to 16x16, the Depth Coordinate X offset needs to be aligned						
		to the 16x16 pixel block.						
6	31:21	Render Target View Extent						
		Format: U11						
		Range: SURFTYPE 1D/2D: same value as Depth field						
		Range: SURFTYPE 3D: [0, 2047] to indicate extent of [1, 2048]						
		For 3D Surfaces:						
		This field indicates the extent of the accessible 'R' coordinates minus 1 on the LOD currently being rendered to.						



	3DSTATE_DEPTH_BUF	FER					
	For 1D and 2D Surfaces: This field must be set to the same value as the Depth field For Other Surfaces: This field is ignored.	d.					
20:0 Reserved							
	Format:	MBZ					



	3	JDSTATE	_DE	PTH_ST	ENCI	L_ST <i>i</i>	41	FE_POINTERS	
Source:		RenderC	S						
Length E	Bias:	2							
Set up t	he poir:	nter to the Dept	er to the Depth Stencil state.						
DWord	Bit				Dese	cription			
0	31:29	Command Type							
		Default Value:				3h	n GF	FXPIPE	
		Format:				Op	рСс	ode	
	28:27	Command Sub	Туре						
		Default Value:		3h GFXPI	IPE	_3D			
		Format:				OpCode			
	26:24	3D Command	Opcod	e					
		Default Value:			0h 3DSTATE_PIPELINED				
		Format:			OpCode	ode			
	23:16	3D Command	3D Command Sub Opcode						
		Default Value: 25h 3DSTAT			_DEPTH_S	TENCIL_S	STA	TE_POINTERS	
		Format:		OpCode					
	15:8	Reserved							
		Format:				MBZ			
	7:0	DWord Length	1						
		Default Value:			0h DWC	RD_COU	JNT	_n	
		Format:			=n				
1	31:6	Pointer to DEP	TH_ST	ENCIL_STATE					
		Format:	Dynam	icStateOffset[3	1:6]DEPTH	DEPTH_STENCIL_STATE			
		Specifies the 6	54-byte	aligned offse	et of the	DEPTH_S	STE	NCIL_STATE. This offset is relative	
		to the Dynam	ic Stat	e Base Addre:	ess				
	5:1	Reserved							
		Format:						MBZ	
	0	Reserved							
		Format:				MB0			



		3DSTAT	'E_DF	RAWING	G_REC	FANGLE			
Source:		RenderCS							
Length E	Bias:	2							
The 3DS	TATE_C	RAWING_RECTANGLE of	ommand	is used to set	the 3D drav	wing rectangle and r	elated sta	te.	
DWord	Bit			Des	cription				
0	31:29	Command Type							
		Default Value:			3h G	FXPIPE			
		Format:			OpC	ode			
	28:27	Command SubType							
		Default Value:			3h GFXPIPI	E_3D			
		Format:			OpCode				
	26:24	3D Command Opcode	•	1					
		Default Value:	1h 3DSTATE_	NONPIPELI	NED				
		Format:	OpCode						
	23:16	3D Command Sub Op	code						
		Default Value:	00h	3DSTATE_DRA	WING_REC	TANGLE			
_		Format:	OpC	ode					
	15:14	Reserved				-			
		Format: MBZ							
	13:8	Reserved							
		Format:			MBZ				
	7:0	DWord Length							
		Default Value:		2h Exclud	2h Excludes DWord (0,1)				
		Format:		=n Total L	=n Total Length - 2				
1	31:16	Clipped Drawing Rect	angle Y N	Vin					
		Format: U16 in Pix	els from	Color Buffer o	rigin (upper	· left corner)			
		Specifies Ymin value of	(inclusive	e) intersection (of Drawing	rectangle with the C	olor (Dest	ination)	
		Value	J. FIXEIS W		Name	IT THIN WILDE CIPPE	u out.		
		[0,16383]	Device ia	nores bits 31:3	0				
		[0]20000]	_ = = : : : ; ;		<u> </u>]		
				Programming	g Notes				
		This value can be large	r than Cli	pped Drawing	Rectangle	Y Max. If Ymin>Yma:	x, the		
		clipped drawing rectangle is null, all polygons are discarded. If Ymin==Ymax, the							
	15.0	Clipped drawing rectar				1.			
	15:0	Eormat: UI16 in Pix	angle X r	viin Color Buffer o	rigin (upper	left corner)			
		Specifies Xmin value of	(inclusive	e) intersection	of Drawing	rectangle with the C	olor (Dest	ination)	
		Buffer, used for clipping	j. Pixels w	vith X coordina	tes less tha	n Xmin will be clippe	ed out.		
		Value Name							



		3DSTA	TE_DRAWING_RECTANGLE								
		[0,16383]	Device ignores bits 15:14								
		Programming Notes									
		clipped drawing rect	ger than Clipped Drawing Rectangle X Max. If Xmin>Xmax, angle is null all polygons are discarded. If Xmin==Xmax, th	the							
		clipped drawing rect	angle is 1 pixel wide in the X direction.								
2	31:16	Clipped Drawing Re	ctangle Y Max								
		Format: U16 in I	Pixels from Color Buffer origin (upper left corner)								
		Specifies Ymax value Buffer, used for clippi	pecifies Ymax value of (inclusive) intersection of Drawing rectangle with the Color (Destination) suffer, used for clipping. Pixels with coordinates greater than Ymax will be clipped out.								
		Value	Name								
		[0,16383]	Device ignores bits 31:30								
			Programming Notes								
		This value can be less	s than Clipped Drawing Rectangle Y Min If Ymax <ymin td="" th<=""><td>e</td></ymin>	e							
		clipped drawing rect	angle is null, all polygons are discarded. If Ymin==Ymax, th	ne							
		clipped drawing rectangle is 1 pixel wide in the Y direction.									
	15:0	Clipped Drawing Rectangle X Max									
		Format: U16 in I	Pixels from Color Buffer origin (upper left corner)	Less (Destination)							
		Buffer, used for clippi	of (inclusive) intersection of Drawing rectangle with the Co ng. Pixels with coordinates greater than Xmax will be clippe	ed out.							
		Value	Name								
		[0,16383]	Device ignores bits 15:14								
			Programming Notos								
		This value can be los	than Clipped Drawing Postangle V Min If Ymay Wing th								
		clipped drawing rect	angle is null, all polygons are discarded. If Xmin==Xmax, th	ne							
		clipped drawing rect	angle is 1 pixel wide in the X direction.								
3	31:16	Drawing Rectangle (Drigin Y								
		Format: S15 in F	ixels from Color Buffer origin (upper left corner).								
		Description									
		Range: [-16384,1638] (Bit 31 should be a sign extension)								
		Specifies Y origin of Drawing Rectangle (in whole pixels) relative to origin of the Color									
		Buffer, used to map i	ncoming (Draw Rectangle-relative) vertex positions to the	Color							
	15.0	Бипеr space.									
	15:0	Format: S15 in E	Jrigin X								
			Description								
		Range: [-16384,1638	3] (Bit 15 should be a sign extension)								



3DSTATE_DRAWING_RECTANGLE	
Specifies X origin of Drawing Rectangle (in whole pixels) relative to origin of the Color Buffer, used to map incoming (Draw Rectangle-relative) vertex positions to the Color Buffer space.	



				3DS	ΤΑΤΕ	_DS				
Source:		Rei	nderCS							
Length B	lias:	2								
The state	e used	by DS is d	efined with	this inline state	packet					
DWord	Bit	-	Description							
0	31:29	Comman	d Type							
		Default Value:				3h GFXPIPE				
		Format:					OpCode			
	28:27	Comman	Command SubType							
		Default \	/alue:			3h GF	XPIPE_3D			
		Format:				ОрСо	de			
	26:24	3D Comr	mand Opco	de						
		Default \		0h 3DS ⁻	TATE_PI	PELINED				
		Format:			OpCode	e				
	23:16	3D Comr	nand Sub C	pcode						
		Default Value:				Dh 3DS	TATE_DS			
		Format:			C	OpCode				
	15:8	Reserved	Reserved							
		Format: MBZ								
	7:0	DWord L	.ength							
		Default \	/alue:	4h Excludes DWord (0,1)			ord (0,1)			
		Format:			=n Total	=n Total Length - 2				
1	31:6	Kernel Start Pointer								
		Format: InstructionBaseOffset[31:6]Kernel								
		This field specifies the starting location of the kernel program run by threads spawned by this FF								
		It is specified as a 64-byte-granular offset from the Instruction Base Address.								
		This field is ignored if DS Function Enable is DISABLED.								
	5:0	Reserved								
	24	Format.	<u> </u>				IVIBZ.			
2	31	Single D	omain Poin	t Dispatch	o aratad T					
		This field	l can ha usa	d to force single		ype point S	IMD4x2 DS threads			
		Value	Name		uomam	point 5	Description			
		0h	Multiple	Dual domain p	oint SIM	D4x2 th	read dispatches are allowed.			
		1h	Single	Single domain	point SIN	ИD4x2 t	hread dispatches are forced.			
	30	Vector N	Jask Enable		•		· · · · · · · · · · · · · · · · · · ·			
	20	Format:		U1 Enun	nerated 1	уре				
		When SP	F=0, Vector	Mask Enable (VI	ME) spec	ifies wh	ich mask to use to initialize the initial			

Г



			3D	STATE	_D	S			
	channel e enables.	enables. W	hen SPF=1, VN	1E specifies	whic	h mask i	to use to generate execution	ר channel	
	Value	Name			Description				
	0h	Dmask	Channels are	enabled bas	sed o	on the d	ispatch mask		
	1h	Vmask	Channels are	enabled bas	sed o	on the ve	ector mask		
29:27	Sampler	Count	-						
	Format:						U3		
	Specifies associate This field	how many d sampler l is ignored	samplers (in n state entries. d if DS Functior	nultiples of 4	4) th DISAI	e kernel BLED.	l uses. Used only for prefetch	ning the	
	Value		Name	Description					
	0h	No Sam	plers	no sample	no samplers used				
	1h	1-4 Sam	plers	between 1	and	4 samp	lers used		
	2h	5-8 Sam	5-8 Samplers		and	8 samp	lers used		
	3h	9-12 Samplers		between 9	and	12 sam	plers used		
	4h	13-16 Sa	between 1	between 13 and 16 samplers used					
26	Reserved	1							
	Format:	Format:			MBZ				
25:18	Binding Table Entry Count								
	Format: U8					U8			
	Specifies binding t Note: For zero to av This field	how man able entrie kernels us void prefet l is ignored	y binding table as and associate ing a large nur cching too man d if DS Functior	e entries the ed surface s mber of bind by entries an n Enable is E	kerr tate. ding d th DISAI	table er rashing BLED.	Used only for prefetching on tries, it may be wise to set t the state cache.	f the his field to	
			Value				Name		
	[0,255]								
17	Reserved	Reserved							
	Format:						MBZ		
16	Floating	Point Mo	de						
	Format:		U1 En	numerated T	ype				
	Specifies Function	the initial Enable is [floating point r DISABLED.	mode used l	oy th	ie dispat	tched thread. This field is igr	nored if DS	
	Va	lue	Nam	е			Description		
	0h		IEEE-754		Use	IEEE-75	4 Rules		
	1h		Alternate		Use	alternat	te rules		
15	Reserved	1							
	Format:						MBZ		
14	Reserved	1							



	3DSTATE_DS							
		Format:		MBZ				
	13	Illegal Opcode Exception Enable						
		Format:	E	nable				
		This bit gets loade Execution Environ	This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.This field is ignored if DS Function Enable is DISABLED.					
	12:8 Reserved							
		Format:		MBZ				
	7	Software Excepti	on Enable					
		Format:	E	nable				
		This bit gets loade	d into EU CR0.1[13] (note the bit #	difference). See Exceptions and ISA				
		Execution Environment. This field is ignored if DS Function Enable is DISABLED.						
	6:0	Reserved						
		Format:		MBZ				
3	31:10	Scratch Space Ba	se Offset					
		Format:	GeneralStateOffset[31:10]ScratchSp	pace				
		aligned offset from the General State Base Address. If required, each thread spawned by this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space. The computed offset of the thread-specific portion will be passed in the thread payload as Scratch Space Offset. The thread is expected to utilize "stateless" DataPort read/write requests to access scratch space, where the DataPort will cause the General State Base Address to be added to the offset passed in the request header. This field is ignored if DS Function Enable is DISABLED.						
	9:4	Reserved						
		Format:		MBZ				
	3:0	Per-Thread Scratch Space						
		Format:	U4 power of 2 Bytes over 1K Byte	s				
		Specifies the amount of scratch space to be allocated to each thread spawned by this FF unit. The driver must allocate enough contiguous scratch space, starting at the Scratch Space Base Pointer, to ensure that the Maximum Number of Threads can each get Per-Thread Scratch Space size without exceeding the driver-allocated scratch space. This field is ignored if DS Function Enable is DISABLED.						
		Value		Name				
		[0,11]	indicating [1K Bytes, 2M Bytes]					
			Duo auo minari	Natas				
		This amount is an	Programming I	notes				
		altered by the ker will ignore it.	mel) to the Data Port in any scratch	space access messages, but the Data Port				



			3DSTATE	DS					
4	31:25	Reserved							
		Format:		MBZ					
	24:20	Dispatch GRF Start R	Dispatch GRF Start Register For URB Data						
		Format:	-	U	5				
		Specifies the starting (thread payload. This fi	Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. This field is ignored if DS Function Enable is DISABLED.						
		Value		Name					
		[0,31]	indicating GRF [R0,R31]						
	19:18	Reserved							
		Format:		MBZ					
	17:11	Patch URB Entry Rea	d Length						
		Format:		U	7				
		Specifies how much dathe DS thread payload	Specifies how much data (in 256-bit units) is to be read from the Patch URB entry and passed in the DS thread payload. This field is ignored if DS Function Enable is DISABLED.						
		١	/alue		Name				
		[0, 64]							
	10	Reserved							
		Format:		MBZ					
		Specifies the offset (in 256-bit units) at which Patch URB data is to be read from the URB before being included in the thread payload. This field is ignored if DS Function Enable is DISABLED.							
		\	/alue		Name				
		[0, 63]							
	3:0	Reserved							
		Format:		MBZ					
5	31:25	Maximum Number o	f Threads		1				
		Format:	U7-1 Thread Cour	nt					
		Specifies the maximum number of simultaneous DS threads allowed to be active. Used to avoid using up the scratch space. Programming the value of the max threads over the number of threads based off number of threads supported in the execution units may improve performance since the architecture allows threads to be buffered between the check for max threads and the actual dispatch into the EU. Programming the max values to a number less than the number of threads supported in the execution units may reduce performance. This field is ignored if DS Function Enable is DISABLED.							
		Value Name		Description					
		[0,15]	indicating thread cour	nt of [1,16]					
	24:21	Reserved							
		Format:		MBZ					
	20:11	Reserved							
	10	Statistics Enable							



	3DSTATE_DS						
	Format:EnableIf ENABLED, this FF unit will engage in statistics gathering. If DISABLED, statisticsinformation associated with this FF stage will be left unchanged. This field is ignored ifDS Function Enable is DISABLED.						
9:3	Reserved						
2	Compute W Coordinate Enable						
	Format: Enable						
	If ENABLED, the DS unit will (for each domain point) compute $W = 1 - (U + V)$ and pass the result as a floating point value in the DS thread payload. If DISABLED, 0.0 will be passed. This field must only be ENABLED for the tessellation of TRI domains, where UVW coordinates are required. This field must be DISABLED for other domains (as they only require UV coordinates) otherwise the computed W coordinate is UNDEFINED. This field is ignored if DS Function Enable is DISABLED.						
1	DS Cache Disable						
	Format: Disable						
	This bit controls the operation of the DS Cache. This field is ignored if DS Function Enable is DISABLED. If the DS Cache is DISABLED and the DS Function is ENABLED, the DS Cache is not used and all incoming domain points will be passed to DS threads. If the DS Cache is ENABLED and the DS Function is ENABLED, incoming domain points that do not hit in the DS Cache will be passed to DS threads. The DS Cache is invalidated whenever the DS Cache becomes DISABLED , whenever the DS Function Enable toggles, and between patches.						
0	DS Function Enable						
	Format: Enable						
If ENABLED, DS threads will be spawned to process incoming domain points whi cache. If DISABLED, the DS stage goes into pass-through mode and performs no speci This field is always used.							
	Programming Notes						
	The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.						



	3DSTATE_DS								
Source:		R	enderCS						
Length E	Bias:	2							
The state	The state used by DS is defined with this inline state packet								
DWord	Bit	Description							
0	31:29	O Command Type							
		Defaul	t Value:				3h (GFXPIPE	
		Format	t:				Opt	Code	
	28:27	Comma	and SubT	уре					
		Defaul	t Value:			3h	GFXPIF	PE_3D	
		Format	t:			Op	Code		
	26:24	3D Con	nmand O	pcode	1				
		Defaul	t Value:		0h 3D	STATE_	PIPELI	NED	
		Format	t:		OpCo	de			
	23:16	3D Con	nmand Su	ıb Opcode					
		Defaul	t Value:			1Dh 3I	1Dh 3DSTATE_DS		
		Format	t:			ОрСос	pCode		
	15:8	Reserve	ed						
		Format	t:					MBZ	
	7:0	DWord	Length						
		Defaul	t Value:		4h Excl	udes D	Word	(0,1)	
		Format	t:		=n Tot	al Leng	jth - 2		
1	31:6	Kernel	Start Poi	nter					
		Format	t: 	InstructionBase	Offset[3	1:6]Ker	rnel		
		unit. It i	a specifie	s the starting location as a 64-byte-grani	on of th ular offs	ie kerne set fror	ei prog n the I	ram run by threads spawned by this FF nstruction Base Address.	
		This fie	ld is igno	red if DS Function E	nable is	DISAB	BLED.		
	5:0	Reserve	ed						
		Format	t:					MBZ	
2	31	Single	Program	Flow (SPF)					
		Format	t: 	U1 Enun	nerated	Туре	: 4	have a single are super flow (CIMDaura	
		Specifies the initial condition of the kernel program as either a single program flow (SIMDnxm with $m = 1$) or as multiple program flows (SIMDnxm with $m > 1$). See CR0 description in ISA						mer a single program flow (SIMD1xm $m > 1$). See CR0 description in ISA	
		Execution Environment.							
Value Name Description						iption			
		0h	Multiple	Multiple Program F normal (SIMD4x2)	lows (1 mode)	- or 2-	vertex	threads spawned, operating under	
		1h	Single	Single Program Flo mode)	w (only	1-vert	ex thre	eads spawned, operating under SPF EU	



			3D	ST/	ATE_DS		
30	Vector Ma	isk Enable (VME)				
	Format: U1 Enumerated Type						
	When SPF=0, VME specifies which mask to use to initialize the initial channel enables. When						
	SPF=1, VME specifies which mask to use to generate execution channel enables.						
	Value	Name	Description				
	0h	Dmask	Channels a	Channels are enabled based on the dispatch mask			
	1h	Vmask	Channels are enabled based on the vector mask				
29:27	Sampler C	pler Count					
	Format:				U3		
	associated sampler state entries. This field is ignored if DS Function Enable is DISABLED						
	Value		Name		Description		
	0h	No Samp	lers		no samplers used		
	1h	1-4 Samp	lers		between 1 and 4 samplers used		
	2h	5-8 Samp	lers		between 5 and 8 samplers used		
	3h	9-12 Samplers			between 9 and 12 samplers used		
	4h	13-16 Sar	between 13 and 16 samplers used				
26	Reserved				·		
	Format:				MBZ		
25:18	Binding Ta When HW	able Entry (Generated	Count Binding Ta	ble i	s disabled:		
	Specifies ł binding tal	now many b ole entries a	inding table nd associate	entri ed sur	es the kernel uses. Used only for prefetching of the rface state.		
	This field i	s ianored if	DS Functio	n Ena	able is DISABLED.		
		g		P	rogramming Notes		
	For kernel to avoid p	s using a lar prefetching t	rge number :oo many en	of bir tries a	nding table entries, it may be wise to set this field to zero and thrashing the state cache.		
17	Thread Pr	iority					
	Format:		U1 En	umer	rated Type		
	Specifies the priority of the thread for dispatch: This field is ignored if DS Function Enable is DISABLED.						
		Value			Name		
	0h		Nor	mal F	Priority		
	1h		High Priority				
16	Floating P	oint Mode					
	Format:		U1 En	umer	rated Type		
Specifies the initial floating point mode use This field is ignored if DS Function Enable is				used by the dispatched thread. ble is DISABLED.			



	3DSTATE_DS								
		Value	Name	Desc	ription				
		0h	Dh IEEE-754						
		1h	Alternate	Use alternate rules					
	15:14	Reserved							
		Format: MBZ							
	13	Illegal Opcode Ex	ception Enable	·					
		Format:	•	Enable					
		This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment. This field is ignored if DS Function Enable is DISABLED.							
	12:8	Reserved							
		Format:		MBZ					
	7	Software Exception	on Enable						
		Format:		Enable					
		This bit gets loade Execution Environr This field is ignore	ets loaded into EU CR0.1[13] (note the bit # difference). See Exceptions and ISA Environment. I is ignored if DS Function Enable is DISABLED.						
	6:0	Reserved							
		Format:		MBZ					
3	31:10	Scratch Space Bas	se Offset						
		Format:	GeneralStateOffset[31:10]S	ratchSpace					
		Specifies the starting location of the scratch space area allocated to this FF unit as a 1K-byte aligned offset from the General State Base Address. If required, each thread spawned by this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space. The computed offset of the thread-specific portion will be passed in the thread payload as Scratch Space Offset. The thread is expected to utilize stateless DataPort read/write requests to access scratch space, where the DataPort will cause the General State Base Address to be added to the offset passed in the request header. This field is ignored if DS Function Enable is DISABLED.							
	9:4	Reserved			1				
		Format:		MBZ					
]								
		Format:	U4 power of 2 Bytes over	1K Bytes					
		The driver must al Pointer, to ensure size without excee This field is ignore	Int of scratch space to be a locate enough contiguous that the Maximum Number ding the driver-allocated sc ed if DS Function Enable is I	ocated to each thread s cratch space, starting at of Threads can each get atch space. ISABLED.	pawned by this FF unit. the Scratch Space Base Per-Thread Scratch Space				



	3DSTATE_DS							
		[0,11]	ind	icating [1K Bytes, 2M Byte	s]			
				Programm	ning Note	25		
		altered by the kernel) to the Data Port in any scratch space access messages, but the Data Port will ignore it.					oatim (if not It the Data Port	
4	31:25	Reserved	Reserved					
		Format:				MBZ		
	24:20	Dispatch GRF	Start R	egister for URB Data				
		Format:				U5		
		Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload.						
		Value			N	lame		
		[0,31]		indicating GRF [R0,R31]				
	19:18	Reserved						
		Format: MBZ						
	17:11	Patch URB Entry Read Length						
		Format:				U7		
		Specifies how the DS thread This field is ig	much da payload nored if	ata (in 256-bit units) is to l DS Function Enable is DIS	pe read fro ABLED.	om the Patch URB entr	y and passed in	
			V	alue		Name		
		[0,64]						
	10	Reserved				1		
		Format:				MBZ		
	9:4	Patch URB En Specifies the o being included This field is ig	RB Entry Read Offset s the offset (in 256-bit units) at which Patch URB data is to be read from the URB before cluded in the thread payload. d is ignored if DS Function Enable is DISABLED.					
			V	/alue		Name		
		[0, 63]						
	3:0	Reserved				1		
		Format:				MBZ		
5	31:25	Maximum Nu	mber o	f Threads			1	
		Format:	. I	J7-1 representing thread of	count			
		Specifies the n using up the s This field is ig	naximun cratch sp nored if	n number of simultaneous bace, or to avoid potential DS Function Enable is DIS	DS thread deadlock ABLED.	ds allowed to be active	e. Used to avoid	
		Value		Nan	ne			
		[0,15] indicating thread count of [1,16]						



3DSTATE_DS

		_				
24:11	Reserved					
	Format:		MBZ			
10	Statistics Enable					
	Format:	Enabl	e			
	If ENABLED, this FF unit will engage in statistics ga associated with this FF stage will be left unchange	thering d.	. If DISABLED, statistics information			
	This field is ignored if DS Function Enable is DISA	BLED.				
9:3	Reserved					
	Format:		MBZ			
2	Compute W Coordinate Enable					
	Format:	Enabl	e			
	If ENABLED, the DS unit will (for each domain point) compute $W = 1 (U + V)$ and pass the result as a floating point value in the DS thread payload. If DISABLED, 0.0 wil be passed. This field must only be ENABLED for the tessellation of TRI domains, where UVW coordinates are required. This field must be DISABLED for other domains (as they only require UV coordinates) otherwise the computed W coordinate is UNDEFINED. This field is ignored if DS Function Enable is DISABLED.					
1	DS Cache Disable					
	Format:	Disable				
	This bit controls the operation of the DS Cache. This field is ignored if DS Function Enable is DISABLED. If the DS Cache is DISABLED and the DS Function is ENABLED, the DS Cache is not used and all incoming domain points will be passed to DS threads. If the DS Cache is ENABLED and the DS Function is ENABLED, incoming domain points that do not hit in the DS Cache will be passed to DS threads. The DS Cache is invalidated whenever the DS Cache becomes DISABLED , whenever the DS Function Enable toggles, and between patches.					
0	DS Function Enable					
	Format:	Enabl	e			
	If ENABLED, DS threads will be spawned to process incoming domain points which miss in the DS cache. If DISABLED, the DS stage goes into pass-through mode and performs no specific processing. This field is always used.					
	Programmi	ng Not	es			
	The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.					



	3DSTATE_GATHER_CONSTANT_HS							
Source:		RenderCS						
Length E	Bias:	2						
This com unit. The	nmand consta	uses the constant buffer bir ant data in these is gathered	nding ta d and pa	ble entries to acked accord	referent	ce constant buffer surface states for HS gather table contained in this command.		
DWord	Bit	5	•	Desc	ription	<u> </u>		
0	31:29	Command Type			-			
		Default Value:			3ł	n GFXPIPE		
		Format:			OpCode			
	28:27	Command SubType						
		Default Value:			3h GFXP	PIPE_3D		
		Format:			OpCode			
	26:24	3D Command Opcode						
		Default Value:		0h 3DSTA	TE_PIPE	LINED		
		Format:		OpCode				
	23:16	3D Command Sub Opcod	le					
		Default Value:	36h 3D	STATE_GAH	ER_CON	ISTANT_HS		
		Format:	OpCod	e				
	15:8	Reserved						
		Format:				MBZ		
	7:0	DWord Length						
		Default Value:	1h	n - 80h Exclud	les DWo	ord (0,1)		
		Format:	= r	า				
		Total Length - 2						
1	31:16	Constant Buffer Valid						
		Format:				U16		
This field specifies which of the 16 constant buffers are used in the push constant ga is set it indicates the corresponding constant buffer is used. If a bit is clear it indicate corresponding constant buffer is not used. If this field is zero it indicate that the gath					used in the push constant gather. If a bit sed. If a bit is clear it indicates the s zero it indicate that the gather buffer is			
		V	alue			Name		
		[0,65535]						
	15:12	5:12 Constant Buffer Binding Table Block						
		Format: U4						
		This field specifies the 16	This field specifies the 16 entry block constant buffer in the binding table. The constant					
		buffer entry block must	be aligr	ned on a 16	entry b	oundary.		
		Value				Name		
		[0,15]						
	11:2	Reserved						



3DSTATE	GATHER	CONSTANT	HS

		Format:		MBZ			
	1	Reserved					
		Format:		MBZ			
	0	Reserved					
		Format:		MBZ			
2	31:23	Reserved					
		Format:		MBZ			
	22:6	Gather Buffer Offset					
		Format: GatherBi	ufferOffset[22:6]				
		This field specifies the offset of the	e gather buffer within th	e Gather Pool.			
			Programming Note	25			
		SW increments the offset by the s generated.	ize of the gather buffer	in 512 bit units for each gather buffer			
	5	Reserved					
		Format:		MBZ			
	4	Reserved					
		Format:		MBZ			
	3	Reserved					
		Format:		MBZ			
	2:0	Reserved					
		Format:		MBZ			
3n	31:24	Constant Buffer Offset for Entry 2n+1					
		Format:	Offset[7:0]				
		This field specifies the Offset in 128-bit units of the 128b entry fetched from the constant buffer for entry 1. Surface Type:ConstantBuffer					
	23:20	Channel Mask for Entry 2n+1					
		Mask:	Mask[3:0]				
		Format:	ConstantBuffer				
		Each bit of this field correspond to the 4 channels of each entry fetched from memory. When the					
		bit is a 1, the corresponding 32-bit value is loaded in FF's push constant buffer. When the bit is a					
		Programming Notes					
		This field may only be zero if it is the last dword of the command packet					
	19.16	Binding Table Index Offset for F	ntry 2n+1	•			
	13.10	Format: Constant Buffer Index	offset [3:0]Surface State	e for ConstantBuffer			
		This field specifies the Binding Tab	le index offset from the	Constant Buffer Binding Table Block			
		starting point in the Binding Table.	. This value is added to	the Constant Buffer Binding Table			
		Block will result in the Binding Tab buffer to be referenced.	ble Index pointing to the	e surface state containing the constant			



	JL	STATE_GAT					
15:8	Constant B	uffer Offset for Entry	2n+0				
	Format:	Offset[7:0	D]ConstantBuffer				
	This field sp	ecifies the Offset in 12	8-bit units of the 128b entry fetched from the constant buffer				
	for entry 2n	for entry 2n+0 (including when On-Die Table Read Enable is set).					
7:4	Channel Ma	ask for Entry 2n+0					
	Mask:		Mask[3:0]				
	Format:		ConstantBuffer				
	Each bit of this field correspond to the 4 channels of each entry fetched from memory. When the						
	bit is a 1, the corresponding 32-bit value is loaded in FF's push constant buffer. When the bit is a						
	0, the corresponding 32-bit value is not loaded. If this field is zero it means the entry is not used.						
3:0	Binding Table Index offset for Entry 2n+0						
	Format:	Constant Buffer Index	offset [3:0]Surface State for ConstantBuffer				
	This field specifies the Binding Table index offset from the Constant Buffer Binding Table Block						
	starting point in the Binding Table. This value is added to the Constant Buffer Binding Table						
	Block will re	Block will result in the Binding Table Index pointing to the surface state containing the constant					
	buffer to be referenced.						



			3D	STA	TE.	GS					
Source:		RenderCS	5								
Length B	Bias:	2									
Controls the GS stage hardware.											
DWord	Bit	-			Des	cription					
0	31:29	Command Typ	Command Type								
		Default Value:				3h G	FXPIPE				
		Format:				OpC	ode				
	28:27	Command Sub	Туре			-					
		Default Value:				3h GFXPIP	E_3D				
		Format:				OpCode					
	26:24	3D Command	Opcode								
		Default Value:		0h 3	DST/	ATE_PIPELIN	IED				
		Format:		ОрС	ode						
	23:16	3D Command	Sub Opcode								
		Default Value:			11	Lh 3DSTATE	_GS				
		Format:			0	pCode					
	15:8	Reserved									
		Format:					MBZ				
	7:0	DWord Length									
		Default Value:		5h Ex	clud	es DWord (0,1)				
		Format:		=n							
1	31:6	Kernel Start Po	ointer								
		Format:	InstructionBa	seOffset	[31:6	jKernel					
		This field specif threads spawne Base Address.	ies the starting loc d by this FF unit. I	ation (1s t is speci	t GE	N4 core inst as a 64-byte	truction) of the kernel program run by e-granular offset from the Instruction				
	5:0	Reserved									
		Format:					MBZ				
2	31	Single Program Flow (SPF) Specifies the initial condition of the kernel program as either a single program flow (SIMDnxm with $m = 1$) or as multiple program flows (SIMDnxm with $m > 1$). See CR0 description in ISA Execution Environment.									
		Value	Name				Description				
	Oh Disable Single Program Flow disabled										
		1h Enable Single Program Flow enabled									
	30	Vector Mask E	nable (VME)								
		Format:	U1 e	numerat	ed ty	/pe					
		When SPF=0, V	ME specifies which	n mask to	o use	to initialize	e the initial channel enables. When				



			3DST	ATE	_GS			
	SPF=1, VN	IE specifi	es which mask to us	e to g	enerate exec	ution cha	nnel enables.	
	Value	Name	9		Des	cription		
	0h	Dmask	Channels are en	abled based on the dispatch mask				
	1h	Vmask	Channels are en	abled	based on th	e vector n	nask	
29:27	Sampler C	Count						
	Format:						U3	
	Specifies h	ecifies how many samplers (in multiples of 4) the geometry shader kernel uses. Used only for						
	prefetching	g the ass	ociated sampler stat	e entr I	ies.		• .•	
	Value		Name			Desc	ription	
	0h	No Sai	nplers	no sa	implers used			
	1h	1-4 Sa	mplers	betw	een 1 and 4	samplers	used	
	2h	5-8 Sa	mplers	betw	een 5 and 8	samplers	used	
	3h	9-12 S	amplers	betw	een 9 and 12	2 sampler	s used	
	4h	13-16	Samplers	betw	een 13 and 2	16 sample	ers used	
	5h-7h	Reserv	ed	Rese	rved			
26	Reserved							
	Format:					MBZ		
25:18	Binding Ta	able Ent	ry Count					
	Format:						U8	
	kernel uses Note: For I zero to avo	s. Used o kernels u pid prefe	nly for prefetching c sing a large number tching too many ent	of the l of bin ries ar	binding table ding table e nd thrashing	e entries a ntries, it n the state	and associated surface nay be wise to set this cache.	e state. s field to
17	Thread Pr Specifies t	iority he priorit	y of the thread for d	ispato	h			
	Valu	е	Nam	Name			Description	
	0h	١	Normal Priority			Normal F	Priority	
	1h	ŀ	ligh Priority			High Pric	ority	
16	Floating P Specifies t	oint Mo he initial	de floating point mode	used	by the dispa	tched three	ead.	
	Valu	Je	Name			D	escription	
	0h		IEEE-754		Use IEEE-75	4 Rules		
	1h alternate				Use alternat	te rules		
15:14	Reserved							
	Format:					MBZ		
13	Illegal Op	code Exc	eption Enable					
	Format:		Enable					
	Double Bu Armed By	uffer :	This bit gets loaded Exceptions and ISA	d into EU CR0.1[12] (note the bit # difference). See A Execution Environment.				e



			3DSTATE_GS							
	12	Reserved								
		Format:		MBZ						
	11	Mask Stack Exception Enable								
		Format:	Enable							
		Double Buffer Armed By:	This bit gets loaded into EU CR0.1 Environment.	[11]. See Exceptions and ISA Execution						
	10:8	Reserved	·							
		Format:		MBZ						
	7	Software Exception E								
		Format:	Enable	e						
		This bit gets loaded in Execution Environmen	This bit gets loaded into EU CR0.1[13] (note the bit $\#$ difference). See Exceptions and ISA Execution Environment.							
	6	Reserved								
		Format:		MBZ						
	5:0	Reserved								
		Format:		MBZ						
3	31:10	Scratch Space Base P	ointer							
		Format:	GeneralStateOffset[31:10]							
		Specifies the location of the scratch space area allocated to this FF unit, specified as a 1KB- granular offset from the General State Base Address. If required, each thread spawned by the unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space.								
	9:4	Reserved								
		Format:		MBZ						
	3:0	Per-Thread Scratch S	pace							
		Format: U4 Power of 2 Bytes over 1K Bytes								
		Specifies the amount of scratch space to be allocated to each thread spawned by this FF unit.The								
		driver must allocate er to ensure that the Max without exceeding the	hough contiguous scratch space, sta kimum Number of Threads can each driver-allocated scratch space.	rting at the Scratch Space Base Pointer, get Per-Thread Scratch Space size						
		Value	Na	me						
		[0,11] ind	icating [1K Bytes, 2M Bytes]							
4	31:29	Reserved								
		Format:		MBZ						
	28:23	Output Vertex Size								
		Format:		U6						
		[0,62] indicating [1,63] 16B units							
		Specifies the size of e	ach vertex stored in the GS output e	entry (following any Control Header						
		data) as a number of	128-bit units (minus one).							



3DSTATE_GS

Programming Notes Programming Restrictions: The vertex size must be programmed as a multiple of 32B units wit the following exception: Rendering is disabled (as per SOL stage state) and the vertex size output by the GS thread is 16B. If rendering is enabled (as per SOL state) the vertex size must be programmed as a multiple of 32B units. In other words, the only time software can program a vertex size with an odd numb of 16B units is when rendering is disabled. 22:17 Output Topology										
Programming Restrictions: The vertex size must be programmed as a multiple of 32B units wit the following exception: Rendering is disabled (as per SOL stage state) and the vertex size output by the GS thread is 16B. If rendering is enabled (as per SOL state) the vertex size must be programmed as a multiple of 32B units. In other words, the only time software can program a vertex size with an odd numb of 16B units is when rendering is disabled. 22:17 Output Topology Format: 3DPrimType This field specifies the topology type (3DPrimType) to be associated with GS-thread output vertices (if any). 16:11 Vertex URB Entry Read Length Specifies the amount of URB data read and passed in the thread payload for each Vertex URB entry, in 256-bit register increments. Programming Restriction:This field must be a non-zero value if Include Vertex Handles is cleared to zero. 10 Include Vertex Handles Format: Boolean This field must be set if Vertex URB handles are included in the payload. These are referred to as "pu model" URB handles, as the thread will use them to read from the URB. Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is uset and GS is enabled, only PATCHLIST topologies may be submitted. Le., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset <td< th=""><th></th><th colspan="9">Programming Notes</th></td<>		Programming Notes								
If rendering is enabled (as per SOL state) the vertex size must be programmed as a multiple of 32B units. In other words, the only time software can program a vertex size with an odd numb of 16B units is when rendering is disabled. 22:17 Output Topology Format: 3DPrimType This field specifies the topology type (3DPrimType) to be associated with GS-thread output vertices (if any). 16:11 Vertex URB Entry Read Length Specifies the amount of URB data read and passed in the thread payload for each Vertex URB entry, in 256-bit register increments. Programming Restriction:This field must be a non-zero value if Include Vertex Handles is cleared to zero. 10 Include Vertex Handles Format: Boolean This field must be set if Vertex URB handles are included in the payload. These are referred to as "pu model" URB handles, as the thread will use them to read from the URB. Programming Notes Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from Buffer Armed 10 Specifies the offset for U		Programming Restrictions: The vertex size must be programmed as a multiple of 32B units with the following exception: Rendering is disabled (as per SOL stage state) and the vertex size output by the GS thread is 16B.								
22:17 Output Topology Format: 3DPrimType This field specifies the topology type (3DPrimType) to be associated with GS-thread output vertices (if any). 16:11 Vertex URB Entry Read Length Specifies the amount of URB data read and passed in the thread payload for each Vertex URB entry, in 256-bit register increments. Programming Restriction:This field must be a non-zero value if Include Vertex Handles is cleared to zero. 10 Include Vertex Handles Format: Boolean If set, all the input Vertex URB handles are included in the payload. These are referred to as "pumodel" URB handles, as the thread will use them to read from the URB. This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read for Buffer Armed 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15		If rendering is er 32B units. In oth of 16B units is w	nabled Ier word Ihen rei	(as per SOL sta ds, the only tim ndering is disat	te) the vo e softwa pled.	ertex size must be re can program a v	programmed as a multiple o ertex size with an odd numb			
Format: 3DPrimType This field specifies the topology type (3DPrimType) to be associated with GS-thread output vertices (if any). 16:11 Vertex URB Entry Read Length Specifies the amount of URB data read and passed in the thread payload for each Vertex URB entry, in 256-bit register increments. Programming Restriction:This field must be a non-zero value if Include Vertex Handles is cleared to zero. 10 Include Vertex Handles Format: Boolean Tf set, all the input Vertex URB handles are included in the payload. These are referred to as "pu model" URB handles, as the thread will use them to read from the URB. Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all By: 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Ivate Name	22:17	Output Topolog	у							
This field specifies the topology type (3DPrimType) to be associated with GS-thread output vertices (if any). 16:11 Vertex URB Entry Read Length Specifies the amount of URB data read and passed in the thread payload for each Vertex URB entry, in 256-bit register increments. Programming Restriction:This field must be a non-zero value if Include Vertex Handles is cleared to zero. 10 Include Vertex Handles Format: Boolean Tf set, all the input Vertex URB handles are included in the payload. These are referred to as "pumodel" URB handles, as the thread will use them to read from the URB. Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHUST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from Buffer Armed the URB before being included in the thread payload. This offset applies to all By: 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] <td></td> <td>Format:</td> <td></td> <td></td> <td>3DPrim</td> <td>Гуре</td> <td></td>		Format:			3DPrim	Гуре				
16:11 Vertex URB Entry Read Length Specifies the amount of URB data read and passed in the thread payload for each Vertex URB entry, in 256-bit register increments. Programming Notes Programming Restriction:This field must be a non-zero value if Include Vertex Handles is cleared to zero. 10 Include Vertex Handles Format: Boolean If set, all the input Vertex URB handles are included in the payload. These are referred to as "pumodel" URB handles, as the thread will use them to read from the URB. Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. Le., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all By: 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] Programming Notes		This field specifie vertices (if any).	es the to	opology type (3	BDPrimTy	/pe) to be associate	ed with GS-thread output			
Programming Notes Programming Restriction:This field must be a non-zero value if Include Vertex Handles is cleared to zero. Include Vertex Handles Format: Boolean If set, all the input Vertex URB handles are included in the payload. These are referred to as "primodel" URB handles, as the thread will use them to read from the URB. Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all Vertex URB entries passed to the thread. 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then Ease public. ColleccT diseate metabeled the id the uit be:	16:11	Vertex URB Entr Specifies the amo entry, in 256-bit i	r y Reac ount of register	d Length URB data read r increments.	and pas	sed in the thread p	ayload for each Vertex URB			
Programming Restriction:This field must be a non-zero value if Include Vertex Handles is cleared to zero. 10 Include Vertex Handles Format: Boolean If set, all the input Vertex URB handles are included in the payload. These are referred to as "pumodel" URB handles, as the thread will use them to read from the URB. Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all Vertex URB entries passed to the thread. 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] If Include Vertex Handles is enabled (pull or hybrid handles case), then Ease Divel to indicating GRF [R0,R15]		Programming Notes								
10 Include Vertex Handles Format: Boolean If set, all the input Vertex URB handles are included in the payload. These are referred to as "promodel" URB handles, as the thread will use them to read from the URB. Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all by: 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] If Include Vertex Handles is enabled (pull or hybrid handles case), then Ear DUIAL ORDECT dispatch me da this field drawidd have		Programming Restriction: This field must be a non-zero value if Include Vertex Handles is cleared to zero.								
Format: Boolean If set, all the input Vertex URB handles are included in the payload. These are referred to as "prmodel" URB handles, as the thread will use them to read from the URB. Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all By: 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] If Include Vertex Handles is enabled (pull or hybrid handles case), then Ear DUAL ORECT dispatch me da this field obculd have	10	Include Vertex H	Handle	S						
If set, all the input Vertex URB handles are included in the payload. These are referred to as "pumodel" URB handles, as the thread will use them to read from the URB. Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all by: 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then Earc DUAL ORIGCT dispatch model (pull or hybrid handles case), then Earc DUAL ORIGCT dispatch model (pull or hybrid handles case), then		Format: Boolean								
Programming Notes This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completel push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all Vertex URB entries passed to the thread. 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then Ease DUAL OR IFCT dignate be meded the field should have		If set, all the input Vertex URB handles are included in the payload. These are referred to as "pul model" URB handles, as the thread will use them to read from the URB.								
This field must be set if Vertex URB Entry Read Length is cleared to zero. When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from Buffer Armed the URB before being included in the thread payload. This offset applies to all By: Vertex URB Entry Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then Ear DUAL ORIECT dimentable mede this field should bati		Programming Notes								
When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completed push all vertex data into the payload. 9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all by: 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] If Include Vertex Handles is enabled (pull or hybrid handles case), then Ear DUAL OR USE media the field should have		This field must be set if Vertex URB Entry Read Length is cleared to zero.								
9:4 Vertex URB Entry Read Offset Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all Vertex URB entries passed to the thread. 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] If Include Vertex Handles is enabled (pull or hybrid handles case), then Ear DUAL ORIECT dispatch media this field should have		When this field is set and GS is enabled, only PATCHLIST topologies may be submitted. I.e., pull-model vertices are only supported for PATCH objects, other object types must completely push all vertex data into the payload.								
Double Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all Vertex URB entries passed to the thread. 3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] If Include Vertex Handles is enabled (pull or hybrid handles case), then Ear PULAL ORIECT dispatch mode this field should have	9:4	Vertex URB Entry Read Offset								
3:0 Dispatch GRF Start Register for URB Data Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then For DUAL ORIECT dispatch mode this field should have		DoubleSpecifies the offset (in 256-bit units) at which Vertex URB data is to be read fromBuffer Armedthe URB before being included in the thread payload. This offset applies to allBy:Vertex URB entries passed to the thread.								
Format: U4 Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then For DUAL ORECT dispatch mode this field should have	3:0	Dispatch GRF St	art Re	gister for URB	Data		1			
Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. Value Name [0,15] indicating GRF [R0,R15] Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then For DUAL OR FECT dispatch mode this field should have		Format:					U4			
Value Name [0,15] indicating GRF [R0,R15] Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then For DUAL ORECT dispatch mode this field should have		Specifies the star thread payload.	ting GF	RF register num	ber for t	he URB portion (Cc	onstant + Vertices) of the			
[0,15] indicating GRF [R0,R15] Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then For DUAL ORIECT dispatch mode this field should have		Value Name								
Programming Notes If Include Vertex Handles is enabled (pull or hybrid handles case), then		[0,15]	i	ndicating GRF [R0,R15]					
If Include Vertex Handles is enabled (pull or hybrid handles case), then				F	Program	ming Notes				
		If Include Verte	ex Han	dles is enable	d (pull c	or hybrid handles	case), then			



					3DST/	ATE_GS				
		(((2*numVerticesPerObject) + 8 - 1)/8) + 1 For SINGLE and DUAL_INSTANCE dispatch modes this field should be: ((numVerticesPerObject +8 - 1)/8) + 1 If Include Primitive ID is set, then add 1 to the value obtained by using the above								
5 33	1:25	Maximum Number of Threads								
		Format: U7-1 thread count								
		Specifie	s the ma	iximu	Im number of simul	taneous threads allowed t	to be active. U	sed to avoid		
		using u Val		atch	space, or to avoid p					
		[0,15]	ir	ndica	ating thread count c	of [1.16]				
	24	Contro	Data Fo	orma	••••••••••••••••••••••••••••••••••••••	. [_,]				
	~ 1	Format	·······································	/////			U1			
		This fiel	d specifie	es th	e format of the con	trol data header (if any).				
		Value	Name	e		Description				
		0h	GSCTL_C	CUT	The control data he	eader contains cut bits.				
		1h	GSCTL_S	SID	The control data header contains StreamID bits Output Topology must be set to POINTLIST, or behavior is UNDEFINED.					
23	3:20	Contro	rol Data Header Size							
		Format	:				U4			
	-	The valu Softwar maixum report r size is z	e nust e num num nore out ero, by d	ber of put verting	there is no control e that the Control D of vertices output by vertices than can be tion neither cut nor	data header, and Control Data Header Size is sufficie y the GS thread. It is UND accomodated in a non-ze StreamID bits are defined	Data Format is nt to accomm EFINED for a G ero-sized head	ignored. odate the S thread to er. (If the header		
				Val	ue		Name			
		[0,8]				32B units				
19	9:15	Instanc Format	e Contro	bl	U5-1 in #	instances				
		[0,31] i	ndicating	j [1 ,3	2] instances					
		Specific docum Instanc DUAL_I Instanc DUAL_ mode.	es the nu ent uses eCount> INSTANC eCount= OBJECT r DUAL_IN	number of instances (minus one) for each input object. To avoid confusion, this es the term "InstanceCount" to refer to InstanceControl+1, with a range of [1,32]If t>1, DUAL_OBJECT mode is invalid. Software will likely want to use NCE mode for higher performance, but SINGLE mode is also supported. When t=1 (one instance per object) software can decide which dispatch mode to use. If mode would likely be the best choice for performance, followed by SINGLE INSTANCE mode is not recommended but is supported.						
14	4:13	Default	Stream	ID			L			
		Format					U2			
		When this field	he GS is e d specifie	enab es the	led, unless the GS c e default StreamID a	output entry contains Strea associated with any GS-th	amID bits in th read output ve	e control header, ertices. When the		



				3DSTATE_GS					
	GS is di	sabled, St	reamID v	vill be output as 0.					
12.11	Dispatch Mode								
12.11	Format: U2								
	This fiel	This field specifies how the GS unit dispatches multiple instances and/or multiple objects							
	Value	Nai	ne	Description					
	0h	SINGLE		Each thread shades a single instance of one object.					
	1h	DUAL_IN	STANCE	Each thread shades possibly two instances of one object. If the InstanceCount is odd, a trailing dispatch of only one instance will be made for each object received. Not recommended if InstanceCount = 1, assuming a kernel optimized for SINGLE or DUAL_OBJECT dispatch would outperform a kernel compiled for DUAL_INSTANCE but only passed one instance. The GS must be allocated at least two URB handles or behavior is UNDEFINED.					
	2h DUAL_OBJECT			Each thread shades one instance of possibly two objects. The GS unit attempt to pair objects together into one dispatch, but under some circumstances only one object may be dispatched (as controlled by the DispatchMask generated by the GS unit). Not valid for objects with more than 16 vertices per object. Not valid if InstanceCount > 1 (more than one instance per object). The GS must be allocated at least two URB handles or behavior is UNDEFINED.					
	3h	Reserved							
10	GS Stat	tistics Ena	ble						
	This bit controls whether GS-unit-specific statistics register(s) can be incremented.								
	Value	ue Name Description							
	0h	Disable	GS_INV	OCATIONS_COUNT and GS_PRIMITIVES_COUNT cannot increment					
	1h	Enable	GS_INV	OCATIONS_COUNT and GS_PRIMITIVES_COUNT can increment					
9:5	GS Invo	ocations I	ncremer	nt Value					
	Format	t:		U5					
	Specifies how much to increment the GS_INVOCATIONS_COUNT for each instance of each object. This control is provided to allow software to process multiple instances (from an API POV in a single kernel invocation. In SINGLE dispatch mode, the counter will increment by this value for each dispatch (as it's only one instance of one object). In DUAL_INSTANCE mode, the counter will be incremented by the value if only one instance is included in the dispatch (i.e., the last odd instance), otherwise the counter will be incremented by the value. In DUAL_OBJECT dispatch mode, the counter will be incremented by the value if only one object), otherwise the counter will be incremented by the value if only one object is included in the dispatch (i.e., a forced dispatch of one object), otherwise the counter will be incremented by twice this value.								
	V	alue		Name					
	[0,31]		indicati	ng an increment of [1,32]					
4	Include	Primitiv	e ID						
	Format	t:		Boolean					
	If set, R	1 of the p	ayload is	written with Primitive ID value(s). If clear, these Primitive ID values are					



	30	STATE_GS
	not included in the payload R1.	
3	Hint	
	Format:	
	This state bit is simply passed in (impact on hardware operation.	S thread payloads for use by the GS kernel - it has no other
2	Reorder Enable	
	Format:	Enable
	This bit controls whether the GS of thread payload. If ENABLED, the of originating from TRISTRIP topolo TRISTRIP_REV topologies. (Note to numbered). With respect to the P TRISTRIP when the vertices are not (regardless of whether a TRISTRIP TRISTRIP/TRISTRIP_REV vertices are received from the pipeline. The G above) so that the GS thread can to account for the non-reordering	unit reorders TRISTRIP/TRISTRIP_REV vertices passed in the GS GS unit will reorder the vertices for "odd-numbered" triangles gies and "even-numbered" triangles originating from hat the first triangle is considered "triangle 0", which is even- rimType passed in the GS thread payload, the GS unit passes of reordered, and TRISTRIP_REV when the vertices are reordered or TRISTRIP_REV topology was being processed)If DISABLED, are not reordered, and always passed in the order they are S unit will still toggle PrimType on alternating (as described perform the reordering internally (or do whatever is necessary g of its input).
1	Discard Adjacency	
	Format:	Enable
	When set, adjacent vertices w adjacency are processed. Inst same fashion as the without-a bit whenever a GS kernel is us both with-adjacency/without- the pipeline (via 3DPRIMITIVE and present the GS thread with will be passed to the GS thread should only clear this bit whe E.g., if the GS kernel is compile this bit. Software should also PATCHLIST_n object (which d	<u>ill not be passed</u> in the GS payload when objects with ead, only the non-adjacent vertices will be passed in the adjacency form of the primitive. Software should set this sed that <u>does not expect</u> adjacent vertices. This allows adjacency variants of the primitive to be submitted to .) - the GS unit will silently discard any adjacent vertices th only the internal object. When clear, adjacent vertices ad, as dictated by the incoming primitive type. Software n a GS kernel is used that does expect adjacent vertices. ed to expect a TRIANGLE_ADJ object, software must clear clear this bit if the GS kernel expects a POINT or on't have with-adjacency variants).
	The only hardware assistance a primitive when operating w of the object. (E.g., when the software should set this bit ju Note that the GS unit is other GS kernel. It is up to software 3DPRIMITIVE) is otherwise co (E.g., if the GS kernel expects	is to allow the submission of a with-adjacency variant of ith a GS kernel that expects the without-adjacency variant GS kernel is compiled to expect a TRIANGLE object, st in case a TRILIST_ADJ is submitted to the pipeline.) wise not aware of the object type that is expected by the to ensure that the submitted primitive type (in mpatible with the object type expected by the GS kernel. a LINE_ADJ object, only LINELIST_ADJ or LINESTRIP_ADJ



		3D	STATE_GS								
		should be submitted, otherwise the GS kernel will produce unpredictable results.) Also note that it is possible to craft a GS kernel which can accept any object type that's thrown at it by first examining the PrimType passed in the payload and then using this info to correctly interpret the number of vertices passed in the payload.									
	0	GS Enable									
		Format:	Ena	ble							
		Specifies whether the GS stage is	enabled or disabled (p	ass-through).							
6	31	Reserved									
		Format: MBZ									
	30:13	Reserved									
		Format: MBZ									
	12	Reserved									
		Format:		MBZ							
	11:0	Semaphore Handle									
		Format:	URBOffset[17:6]								
		This is the URB offset pointing to the region is 128 DWs(8 - 512b U and/or HS semaphore Dwords in make sure the 3D pipeline is IDLE can be located in an unused area the Push Constant region.	to the first of the GS semaphore DWords in the URB. The size of OURB entries). Software is responsible for allocating combined GS in a single contiguous region of the URB. Software must also OLE prior to allocating or deallocating the region. The semaphores ea within a FF unit's URB fenced region or an unused area within								



			3D	STA	E_GS	5				
Source:		RenderC	5							
Length E	Bias:	2								
Controls the GS stage hardware.										
DWord	Bit				Descript	on				
0	31:29	Command Typ	e							
		Default Value:				3h C	GFXPIPE			
		Format:				ОрС	Code			
	28:27	Command Sub	Туре							
		Default Value:			3h C	FXPIP	E_3D			
		Format:			ОрС	ode				
	26:24	3D Command	Opcode							
		Default Value:		0h 31	DSTATE_F	PIPELIN	NED			
		Format:		OpC	ode					
	23:16	3D Command	Sub Opcode							
		Default Value:			11h 3D	STATE	E_GS			
		Format:			OpCod	е				
	15:8	Reserved								
		Format:					MBZ			
	7:0	DWord Length								
		Default Value:		5h Ex	ludes D\	Vord (0,1)			
		Format:		=n						
1	31:6	Kernel Start Po	ointer							
		Format:	InstructionBa	seOffset	31:6]Kerr	nel				
		This field specif	ies the starting loc	ation (1 st	GEN4 co	re inst	ruction) of the kernel program run by			
		Inreads spawned by this FF unit. It is specified as a 64-byte-granular offset from the Instruction Base Address.								
	5:0	Reserved								
		Format:					MBZ			
2	31	Single Program	n Flow							
		Format:				Enabl	e			
		Specifies the in	itial condition of th	ne kernel	program	as eitl	ner a single program flow (SPF)			
		ארעואונט $m = 1$) or as multiple program flows (SIMDnxm with m > 1). See CRU de in ISA Execution Environment.								
		Value	Name				Description			
	sabled									
		1	Enable	Single Pr	ogram Fl	ow en	able			
	30	Vector Mask E	nable	9.0.1	5					
	50	When SPF=0, V	ector Mask Enable	(VME) sp	ecifies w	hich n	nask to use to initialize the initial			



			3	DST	ATE	E_GS			
	channel en enables.	ables. Wh	en SPF=1,	, VME spe	cifies	which mask to use to generate execution channel			
	Value	Name				Description			
	0h	Dmask	Channe	Channels are enabled based on the dispatch mask					
	1h	Vmask	Channe	els are ena	abled	based on the vector mask			
29:27	Sampler C	ount							
	Format:					U3			
	Specifies h	now man	y sample	ers (in mu	ltiple	es of 4) the geometry shader kernel uses. Used			
	only for prefetching the associated sampler state entries.								
	Value		Name			Description			
	0h	No Sam	plers		no sa	amplers used			
	1h1-4 Samplersbetw					veen 1 and 4 samplers used			
	2h	5-8 Sam	plers		betw	veen 5 and 8 samplers used			
	3h	9-12 Sai	mplers		betw	veen 9 and 12 samplers used			
	4h	13-16 Sa	amplers		betw	veen 13 and 16 samplers used			
	5h-7h	7h Reserved							
26	Reserved								
	Format:					MBZ			
25:18	Binding Ta	ble Entry	Count						
	Format:					U8			
	When Hw	Generat	ed Bindin	ng lable is	s disa				
	binding table entries and associated surface state.								
	Programming Notes								
	For kernels to avoid p	s using a l refetching	arge num too man	ber of bin y entries a	iding and th	table entries, it may be wise to set this field to zero nrashing the state cache.			
17	Thread Pri Specifies th	ority ne priority	of the thr	read for di	ispato	ch			
		Value				Name			
	0h			Normal P	riority	У			
	1h High Priority								
16	Floating Po Specifies th	oint Mod	e oating po	int mode	used	by the dispatched thread.			
	Value Name Description								
	0h	I	EE-754			Use IEEE-754 Rules			
	1h	а	lternate			Use alternate rules			
15:14	Reserved								

Г



				3	DSTATE_GS					
		Format:				MBZ				
	13	Illegal Opcode	Except	ion Ena	ble					
		Format:								
		Double BufferThis bit gets loaded into EU CR0.1[12] (note the bit # differeArmed By:Exceptions and ISA Execution Environment.								
	12	Reserved								
		Format:				MBZ				
	11	Mask Stack Exc	eption	Enable						
		Format:		Enable						
		Double Buffer A By:	Armed	This bit Environ	gets loaded into EU CR0.1	[11]. See Exceptions and ISA Execution				
		This bit gets load	ded int	o EU CR	0.1[11]. See Exceptions and	ISA Execution Environment.				
	10:8	Reserved								
		Format:				MBZ				
	7	Software Excep	tion E	nable						
		Format:			Enable	e				
		This bit gets load	ded int	o EU CR	0.1[13] (note the bit # diffe	rence). See <i>Exceptions</i> and <i>ISA</i>				
		Execution Enviro	nment.							
	6:0	Reserved								
		Format:				MBZ				
3	31:10	Scratch Space E	Base Po	Dinter						
		Format:	ation	Gener	aistateomset[31:10]	a this EE unit spacified as a 1KP				
		granular offset fi unit will be alloc	rom th ated so	e Genera ome por	al State Base Address. If req tion of this space, as specifi	juired, each thread spawned by this FF ied by Per-Thread Scratch Space.				
	9:4	Reserved								
		Format:				MBZ				
	3:0	Per-Thread Scra	atch Sj	oace						
		Format:	U4	power o	of 2 Bytes over 1K Bytes					
		Specifies the amount of scratch space to be allocated to each thread spawned by this FF unit.								
		The driver must allocate enough contiguous scratch space, starting at the Scratch Space Base								
		size without exceeding the driver-allocated scratch space.								
		Value Name Description								
		[0,11]			indicating [1K Bytes, 2M B	ytes]				
4	31:29	Reserved								
		Format:				MBZ				
	28:23	Output Vertex Size								



		503							
	Format: U6								
	[0 63] indicating [1 64] 16B units								
	Specifies the size of each vertex stored in the GS output entry (following any Control Header data) as a number of 128-bit units (minus one).								
		Programming Notes							
	The vertex size m Rendering is disa 16B. If rendering is en	nust be programme bled (as per SOL s abled (as per SOL	ed as a multiple of 32B units with the following exception: tage state) and the vertex size output by the GS thread is state) the vertex size must be programmed as a multiple of						
	of 16B units is wh	sabled.							
22:17	Output Topology	y							
	Format:		3DPrimType						
	This field specifies the topology type (3DPrimType) to be associated with GS-thread output vertices (if any).								
10.11	Manten LIDD Frider	D a sol 1 and solar							
16:11	Vertex URB Entry Specifies the amo entry, in 256-bit re	y Read Length unt of URB data re egister increments	ad and passed in the thread payload <u>for each Vertex URB</u> .						
16:11	Specifies the amo entry, in 256-bit re This field must be	y Read Length unt of URB data re egister increments e a non-zero value	ad and passed in the thread payload <u>for each Vertex URB</u> . Programming Notes if Include Vertex Handles is cleared to zero.						
16:11	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H	y Read Length unt of URB data re egister increments e a non-zero value andles	ead and passed in the thread payload <u>for each Vertex URB</u> . Programming Notes if Include Vertex Handles is cleared to zero.						
16:11	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format:	y Read Length unt of URB data re egister increments e a non-zero value andles	ad and passed in the thread payload <u>for each Vertex URB</u> . Programming Notes if Include Vertex Handles is cleared to zero. Boolean						
16:11	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format: If set, all the input model URB handle	y Read Length unt of URB data re egister increments e a non-zero value andles t Vertex URB handl es, as the thread w	ad and passed in the thread payload <u>for each Vertex URB</u> Programming Notes if Include Vertex Handles is cleared to zero. Boolean es are included in the payload. These are referred to as pull ill use them to read from the URB.						
16:11	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format: If set, all the input model URB handle	y Read Length unt of URB data re egister increments e a non-zero value andles t Vertex URB handl es, as the thread w	aad and passed in the thread payload <u>for each Vertex URB</u> . Programming Notes if Include Vertex Handles is cleared to zero. Boolean es are included in the payload. These are referred to as pull ill use them to read from the URB. Programming Notes						
16:11	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format: If set, all the input model URB handle When set (and th operation is UND This field must be	y Read Length unt of URB data re egister increments e a non-zero value andles t Vertex URB handl es, as the thread w e GS is enabled) , o DEFINED. e set if Vertex URB	ead and passed in the thread payload <u>for each Vertex URB</u> Programming Notes if Include Vertex Handles is cleared to zero. Boolean es are included in the payload. These are referred to as pull ill use them to read from the URB. Programming Notes only PATCHLIST topologies can be issued to the pipeline, or BEntry Read Length is cleared to zero.						
9:4	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format: If set, all the input model URB handle When set (and th operation is UND This field must be Vertex URB Entry	y Read Length unt of URB data re egister increments e a non-zero value andles t Vertex URB handl es, as the thread w be GS is enabled) , o DEFINED. e set if Vertex URB y Read Offset	ead and passed in the thread payload <u>for each Vertex URB</u> Programming Notes if Include Vertex Handles is cleared to zero. Boolean es are included in the payload. These are referred to as pull ill use them to read from the URB. Programming Notes only PATCHLIST topologies can be issued to the pipeline, or BENTY Read Length is cleared to zero.						
16:11 10 9:4	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format: If set, all the input model URB handle When set (and th operation is UND This field must be Vertex URB Entry Double Sp Buffer Armed th By: Vertex	y Read Length unt of URB data re egister increments e a non-zero value andles t Vertex URB handl es, as the thread w DEFINED. e set if Vertex URB y Read Offset pecifies the offset (the URB before bein ertex URB entries p	Programming Notes if Include Vertex Handles is cleared to zero. Boolean es are included in the payload. These are referred to as pull ill use them to read from the URB. Programming Notes only PATCHLIST topologies can be issued to the pipeline, or Bentry Read Length is cleared to zero. (in 256-bit units) at which Vertex URB data is to be read from ng included in the thread payload. This offset applies to all passed to the thread.						
16:11 10 9:4 3:0	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format: If set, all the input model URB handle When set (and the operation is UND This field must be Vertex URB Entry Double Sp Buffer Armed th By: Vetex	y Read Length unt of URB data re egister increments e a non-zero value andles t Vertex URB handl es, as the thread w DEFINED. e set if Vertex URI y Read Offset pecifies the offset (the URB before bein ertex URB entries p art Register for U	Programming Notes if Include Vertex Handles is cleared to zero. Boolean es are included in the payload. These are referred to as pull ill use them to read from the URB. Programming Notes only PATCHLIST topologies can be issued to the pipeline, or Bentry Read Length is cleared to zero. (in 256-bit units) at which Vertex URB data is to be read from ag included in the thread payload. This offset applies to all bassed to the thread. RB Data						
16:11 10 9:4 3:0	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format: If set, all the input model URB handle When set (and th operation is UND This field must be Vertex URB Entry Double Sp Buffer Armed th By: Vertex IRF State Format: Sp	y Read Length unt of URB data re egister increments e a non-zero value andles t Vertex URB handl es, as the thread w be GS is enabled) , DEFINED. e set if Vertex URB opecifies the offset pecifies the offset (be URB before bein ertex URB entries p art Register for UI	ead and passed in the thread payload <u>for each Vertex URB</u> Programming Notes if Include Vertex Handles is cleared to zero. Boolean es are included in the payload. These are referred to as pull ill use them to read from the URB. Programming Notes only PATCHLIST topologies can be issued to the pipeline, or Bentry Read Length is cleared to zero. (in 256-bit units) at which Vertex URB data is to be read from ag included in the thread payload. This offset applies to all bassed to the thread. RB Data U4						
16:11 10 9:4 3:0	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format: If set, all the input model URB handle When set (and the operation is UND This field must be Vertex URB Entry Double Sp Buffer Armed th By: Vetex Vertex URB Entry Specifies the start thread payload. Specifies the start	y Read Length unt of URB data re egister increments e a non-zero value andles t Vertex URB handl es, as the thread w be GS is enabled) , o DEFINED. e set if Vertex URB pecifies the offset pecifies the offset (he URB before bein ertex URB entries p art Register for UI ing GRF register no	Programming Notes if Include Vertex Handles is cleared to zero. Boolean es are included in the payload. These are referred to as pull ill use them to read from the URB. Programming Notes only PATCHLIST topologies can be issued to the pipeline, or Bentry Read Length is cleared to zero. (in 256-bit units) at which Vertex URB data is to be read from ng included in the thread payload. This offset applies to all bassed to the thread. RB Data U4 umber for the URB portion (Constant + Vertices) of the						
16:11 10 9:4 3:0	Vertex URB Entry Specifies the amo entry, in 256-bit re This field must be Include Vertex H Format: If set, all the input model URB handle When set (and the operation is UND This field must be Vertex URB Entry Double Specifies the start Buffer Armed th By: Vetex URB Entry Double Specifies the start thread payload. Value	y Read Length unt of URB data re egister increments e a non-zero value andles t Vertex URB handl es, as the thread w he GS is enabled), of DEFINED. e set if Vertex URB pecifies the offset pecifies the offset (he URB before bein ertex URB entries p art Register for UI ing GRF register no Name	Programming Notes if Include Vertex Handles is cleared to zero. Boolean es are included in the payload. These are referred to as pull ill use them to read from the URB. Programming Notes only PATCHLIST topologies can be issued to the pipeline, or Bentry Read Length is cleared to zero. (in 256-bit units) at which Vertex URB data is to be read from ng included in the thread payload. This offset applies to all bassed to the thread. RB Data U4 umber for the URB portion (Constant + Vertices) of the Description						



3DSTATE_GS											
		Programming Notes									
		If Include Vertex Handles is enabled (pull or hybrid handles case), then For DUAL_OBJECT dispatch mode this field should be: (((2*numVerticesPerObject) + 8 - 1)/8) + 1 For SINGLE and DUAL_INSTANCE dispatch modes this field should be: ((numVerticesPerObject +8 - 1)/8) + 1									
		If Inclu	ide Prin	nitive I	D is set, th	nen add 1	to the value obtained by using the above	'e			
5	31:25	Maxim	um Nu	mber	of Thread	ds					
		Forma	t:			U7					
		Forma	t:			U7-1 thre	ead count				
		Specifies the maximum number of simultaneous threads allowed to be active. Used to avoid using up the scratch space, or to avoid potential deadlock.									
		Va	lue		Name						
		[0,15] indicating three				nread count of [1,16]					
		[0,35]		indica	ating threa	ad count o	of [1,36]				
	24	Contro	l Data	Forma	ət						
		Forma	t:				U1				
		This fie	eld spe	cifies	the form	at of the	control data header (if any).				
		Value	Nar	ne			Description				
		0h	GSCTL	_CUT	The contr	ol data he	eader contains cut bits.				
		1h GSCTL_SID The control data header contains StreamID bits. Output Topology muscless set to POINTLIST, or behavior is UNDEFINED.									
	23:20	Control Data Header Size									
		Format: U4									
		Specifies the number of 32B units of control data header located at the start of the GS URB entry. The value 0 indicates there is no control data header, and Control Data Format is ignored. Software must ensure that the Control Data Header Size is sufficient to accommodate the maixumum number of vertices output by the GS thread. It is UNDEFINED for a GS thread to report more output vertices than can be accomodated in a non-zero-sized header. (If the header size is zero, by definition neither cut nor StreamID bits are defined.									
	Value										
		[0,8]	32B units								
	19:15	Instand	nstance Count								
Format: U5-1 in #instances											
		Specifies the number of instances (minus one) for each input object. To avoid confusion, this document uses the term InstanceCount to refer to InstanceControl+1, with a range of [1,32] If InstanceCount >1, DUAL_OBJECT mode is invalid. Software will likely want to use DUAL_INSTANCE mode for higher performance, but SINGLE mode is also supported. When InstanceCount =1 (one instance per object) software can decide which dispatch mode to use. DUAL_OBJECT mode would likely be the best choice for performance, followed by SINGLE mode. DUAL_INSTANCE mode is not recommended but is supported.									
	14:13	Default StreamID									


				3DSTATE_GS					
	Format	t:		U2					
	When the GS is enabled, unless the GS output entry contains StreamID bits in the control header this field specifies the default StreamID associated with any GS-thread output vertices. When the GS is disabled, StreamID will be output as 0.								
12:11	Dispate	h Mode							
	Format	t:		U2					
	This fiel	d specifie	s how th	e GS unit dispatches multiple instances and/or multiple objects.					
	Value	Nai	me	Description					
	0h	SINGLE		Each thread shades a single instance of one object.					
	1h DUAL_INSTANCE			Each thread shades possibly two instances of one object. If the InstanceCount is odd, a trailing dispatch of only one instance will be made for each object received. Not recommended if InstanceCount = 1, assuming a kernel optimized for SINGLE or DUAL_OBJECT dispatch would outperform a kernel compiled for DUAL_INSTANCE but only passed one instance. The GS must be allocated at least two URB handles or behavior is UNDEFINED.					
	2h	n DUAL_OBJECT		Each thread shades one instance of possibly two objects. The GS unit attempt to pair objects together into one dispatch, but under some circumstances only one object may be dispatched (as controlled by the DispatchMask generated by the GS unit). Not valid for objects with more than 16 vertices per object. Not valid if InstanceCount > 1 (more than one instance per object). The GS must be allocated at least two URB handles or behavior is UNDEFINED.					
	3h	Reserved							
10	GS Stat	tistics Fna	hle						
10	This bit	controls	whether (GS-unit-specific statistics register(s) can be incremented.					
	Value	Name		Description					
	0h	Disable	GS_INV	OCATIONS_COUNT and GS_PRIMITIVES_COUNT cannot increment					
	1h	Enable	GS_INV	OCATIONS_COUNT and GS_PRIMITIVES_COUNT can increment					
9:5	GSInvo	cations I	ncremen	nt Value					
	Format	Format: U5							
	Specifies how much to increment the GS_INVOCATIONS_COUNT for each instance of each object. This control is provided to allow software to process multiple instances (from an API POV) in a single kernel invocation. In SINGLE dispatch mode, the counter will increment by this value for each dispatch (as it's only one instance of one object). In DUAL_INSTANCE mode, the counter will be incremented by the value if only one instance is included in the dispatch (i.e., the last odd instance), otherwise the counter will be incremented by twice this value. In DUAL_OBJECT dispatch mode, the counter will be incremented by the value if only one object is included in the dispatch (i.e., a forced dispatch of one object), otherwise the counter will be								
	V	alue		Name					



		503	TAIL_G	15				
[0,3	31]	indicating an incre	ment of [1,32]]				
4 Incl	ude Primitiv	ve ID						
For	mat:		Во	polean				
If se	t, R1 of the p	payload is written wit	th Primitive ID) value(s).				
IT CIE	ear, these Pri	Imitive ID values are i	not included li	in the payload R1.				
3 Hin	t							
For	mat:			U1				
This imp	This state bit is simply passed in GS thread payloads for use by the GS kernel it has no other impact on hardware operation.							
2 Reo	rder Enable	,						
For	mat:			Enable				
This	bit controls	whether the GS unit	reorders TRIS	STRIP/TRISTRIP_REV vertices passed in the GS				
thre	ad payload.							
If EN	If ENABLED, the GS unit will reorder the vertices for odd-numbered triangles originating from							
TRIS	RIP topolo	gies and even-numbe	ered triangles	s originating from TRISTRIP_REV topologies.				
(INO	With respect to the PrimType passed in the GS thread payload the GS unit passes TRISTRIP who							
VVIT	n respect to	the Primi ype passed		ead payload, the GS unit passes TRISTRIP who				
the	thor a TRICT	<u>not</u> reordered, and Tr DID or TDICTDID DEV/	topology was	when the vertices <u>are</u> reordered (regardless o				
If DI		ISTRIP/TRISTRIP REV	vertices are no	not reordered, and always passed in the order				
they	are received	d from the pipeline T	The GS unit wi	ill still toggle PrimType on alternating (as				
desc	cribed above	e) so that the GS threa	ad can perform	m the reordering internally (or do whatever is				
nece	essary to acc	count for the non-reo	ordering of its	input).				
1 Disc	ard Adjace	ncy						
For	mat:	-		Enable				
Wh	nen set, adia	acent vertices will n	not be passed	d in the GS payload when objects with				
adi	acency are	processed. Instead	only the no	on-adjacent vertices will be passed in the				
Isar	ne fashion	as the without-adia	acency form (of the primitive Software should set this				
bit	whenever a	a GS kernel is used	that does no	of the primitive. Software should set this				
bo	th with_adi	a ob kemens used	iaconcy varia	unts of the primitive to be submitted to				
the	ninolino (v		the CS unit w	will silently discard any adjacent vertices				
line	e pipelille (v	a CC thread with a	une do unit w	will sherify discard any adjacent vertices				
and	a present tr	he GS thread with o	only the inter	mai object. when clear, adjacent vertices				
<u>wii</u>	<u>i be passed</u>	to the GS thread, a	as dictated b	by the incoming primitive type. Software				
sno	ould only cl	ear this bit when a	GS kernel is	used that does expect adjacent vertices.				
	., if the GS I	kernel is compiled t		IRIAN(TEAD) object software must clea				
L.Y			to expect a 1					
this	s bit. Softwa	are should also clea	ar this bit if th	he GS kernel expects a POINT or				
this PA	s bit. Softwa TCHLIST_n	are should also clea object (which don't	to expect a 1 ar this bit if th t have with-a	the GS kernel expects a POINT or adjacency variants).				
this PA The	s bit. Softwa TCHLIST_n e only hard	are should also clea object (which don't ware assistance is t	ar this bit if the transformed of transforme	whe GS kernel expects a POINT or adjacency variants). submission of a with-adjacency variant o				
this PA The a p	s bit. Softwa TCHLIST_n e only hard primitive wh	are should also clea object (which don't ware assistance is t en operating with a	ar this bit if th t have with-a o allow the s a GS kernel t	the GS kernel expects a POINT or adjacency variants). submission of a with-adjacency variant o that expects the without-adjacency variant				



3DSTATE_GS software should set this bit just in case a TRILIST_ADJ is submitted to the pipeline.) Note that the GS unit is otherwise not aware of the object type that is expected by the GS kernel. It is up to software to ensure that the submitted primitive type (in 3DPRIMITIVE) is otherwise compatible with the object type expected by the GS kernel. (E.g., if the GS kernel expects a LINE_ADJ object, only LINELIST_ADJ or LINESTRIP_ADJ should be submitted, otherwise the GS kernel will produce unpredictable results.) Also note that it is possible to craft a GS kernel which can accept any object type that's thrown at it by first examining the PrimType passed in the payload and then using this info to correctly interpret the number of vertices passed in the payload. 0 **GS Enable** Format: Enable Specifies whether the GS stage is enabled or disabled (pass-through). 6 31:12 Reserved Format: MBZ 11:0 Semaphore Handle U12 Handle Format: This is the 512b-aligned URB handle pointing to the first of the GS semaphore DWords in the URB. Software is responsible for statically allocating combined GS and/or HS semaphore Dwords in a single contiguous region of the URB. The semaphores can be located in an unused area within a FF unit's URB fenced region or an unused area within the Push Constant region.



3DSTATE	HIER	DEPTH	BUFFER
	-	-	

Source: RenderCS

Length Bias: 2

This command sets the surface state of the hierarchical depth buffer, delivered as a pipelined state command. However, the state change pipelining isn't completely transparent (see restriction below).

Programming Notes

Restriction: Prior to changing Depth/Stencil Buffer state (i.e., any combination of 3DSTATE_DEPTH_BUFFER, 3DSTATE_CLEAR_PARAMS, 3DSTATE_STENCIL_BUFFER, 3DSTATE_HIER_DEPTH_BUFFER) SW must first issue a pipelined depth stall (PIPE_CONTROL with Depth Stall bit set, followed by a pipelined depth cache flush (PIPE_CONTROL with Depth Flush Bit set, followed by another pipelined depth stall (PIPE_CONTROL with Depth Stall Bit set), unless SW can otherwise guarantee that the pipeline from WM onwards is already flushed (e.g., via a preceding MI_FLUSH).

3DSTATE_HIER_DEPTH_BUFFER must always be programmed in the along with the other Depth/Stencil state commands(i.e. 3DSTATE_DEPTH_BUFFER, 3DSTATE_CLEAR_PARAMS, or 3DSTATE_STENCIL_BUFFER)

DWord	Bit	Description							
0	31:29	Command Ty	ре						
		Default Value	:				3h G	FXPIPE	
		Format:					OpCode		
	28:27	Command Su	bType						
		Default Value	:					E_3D	
		Format:			OpCo	ode			
	26:24	3D Command	Opcode						
		Default Value			0h 3DST	ATE_PI	IPELIN	NED	
		Format:			OpCode				
	23:16	3D Command Sub Opcode							
		Default Value		07h 3D	STATE_HII	ER_DEI	PTH_E	BUFFER	
		Format:		OpCod	e				
	15:8	Reserved							
		Format:						MBZ	
	7:0	Dword Length							
		Format:		=n Tot	al Length	- 2			
		Value			Na	me			
		_1h	Excludes Dwo	ord (0,1)	[Default]				
1	31:29	Reserved						1	
Format: MBZ						MBZ			
	28:25	Hierarchical D	Depth Buffer	Object (Control St	tate			
		Format:	MEMORY	Y_OBJEC	T_CONTR	OL_ST/	ATE		



		3DS	STATE_HIER_DEPTH_BUFFER							
			Description							
		Specifies the m	nemory object control state for the hierarchical depth buffer.							
		This field is not context save and restored by hardware. If this field is programmed to any value other than zero, it must be programmed after the following commands or events:								
		MI_SET_CONTEXT								
		MI_WAIT_FOR_EVENT (Specifically waits on vblank or display flip)								
		Render e	Render engine goes IDLE due to head point equal to tail pointer							
	24:17	Reserved								
		Format: MBZ								
	16:0	Surface Pitch								
		Format:	U17-1 Pitch in Bytes							
		This field specifi	ies the pitch of the hierarchical depth buffer in (#Bytes - 1).							
		Value	Name							
		[127, 3FFFFh]	corresponding to [128B, 128KB] also restricted to a multiple of 128B							
			Programming Notes							
		Since this surface [128B, 128KB].	ce is tiled, the pitch specified must be a multiple of the tile pitch, in the rar	nge						
2	31:0	Surface Base A	ddress							
		Format:	GraphicsAddress[31:0]HierarchicalDepthBuffer							
		This field specifi	ies the starting Dword address of the buffer in mapped Graphics Memory.							
			Programming Notes							
		The Hierarchica	al Depth Buffer can only be mapped to Main Memory (uncached).							



			3D	S1	ΓΑΤ	E_HS						
Source:		Renc	derCS									
Length B	Bias:	2										
Controls	the HS	stage hard	lware.									
DWord	Bit				D	escriptic	on					
0	31:29	Command	Туре									
		Default Va	alue:				3h G	FXPIPE				
		Format: OpCode										
	28:27	Command	SubType									
		Default Va	alue:			3h GF	XPIPI	E_3D				
		Format:				ОрСо	de					
	26:24	3D Comma	3D Command Opcode									
		Default Va	alue:		0h 3D9	STATE_PI	PELIN	IED				
		Format:			ОрСос	le						
	23:16	3D Comma	D Command Sub Opcode									
	Default Value:					1Bh 3DSTATE_HS						
Format:						OpCode						
	15:8	5:8 Reserved										
		Format:						MBZ				
	7:0	DWord Le	ngth						1			
		Format:							=n			
		Value				Jamo						
		5	Excludes DWord (0.1) [Default]									
1	21.20	Percented		0,1)								
T	51.50	Format [.]						MB7				
	20.27	Sampler C										
	29.21	Format:	ount						U3			
		Specifies h	ow many samplers (in	mult	tiples o	f 4) the H	IS ker	nels use	e. Used only	for pre	fetching the	
		associated	sampler state entries.		·						Ū	
		Value	Name				Desc	ription				
		0h	No Samplers	nc	o sampl	ers used						
		1h	1-4 Samplers	be	etween	1 and 4 s	sampl	lers used	b			
		2h	5-8 Samplers	be	etween	5 and 8 s	sampl	lers used	b			
		3h	9-12 Samplers	be	etween	9 and 12	sam	olers use	ed			
4h 13-16 Samplers between 13 and 16 samplers						nplers u	sed					
		5h-7h	Reserved	Re	eserved							
	26	Reserved									1	
		Format:						MBZ				



		3D3	STATE_HS						
25:18	Binding Table	e Entry Count							
	Format:			U8					
	When HW Generated Binding Table is disabled:								
	Specifies how	many binding table	entries the kernel uses. Used	only for prefete	ching of the				
	binding table	entries and associate	ed surface state.						
	Note: For ker	nels using a large nu	mber of binding table entries	, it may be wise	to set this field to				
	zero to avoid	prefetching too many	y entries and thrashing the st	ate cache.					
17	Reserved								
	Format:		MBZ						
16	Floating Poin	t Mode							
	Specifies the initial floating point mode used by the dispatched thread.								
	Value	Name	Description	l .					
	0h	IEEE-754	Use IEEE-754 Rules						
	1h	alternate	Use alternate rules						
15:14	4 Reserved								
	Format:		MBZ						
13	Illegal Opcode Exception Enable								
	This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.								
12	Reserved								
	Format:		MBZ						
11:8	Reserved								
	Format:		MBZ						
7	Software Exc	eption Enable							
	Format:	•	Enable						
	This bit gets lo	paded into EU CR0.1[13] (note the bit # difference)). See Exception	s and ISA				
	Execution Env	ironment.		·					
6:0	Maximum Nu	umber of Threads							
	Format:	U7-	1 thread count						
	Specifies the r	maximum number of	simultaneous threads allowed	d to be active. L	Jsed to avoid				
	using up the s	cratch space. Program	mming the value of the max t	threads over the	e number of				
	since the arch	itecture allows thread	ds supported in the execution ds to be buffered between the	e check for max	threads and the				
	actual dispatc	h into the EU. Progra	mming the max values to a n	umber less thar	the number of				
	threads suppo	orted in the execution	units may reduce performar	nce. Limit is base	ed on max				
	number of HS	URB handles.			_				
	Value		Name						
	Value Name								



				30	DSTATE_HS					
				and must h	Programming Note	es	eld and			
		before ar Graphics	ny subseque Processing	ent pipeline Engine (Co	processing (e.g., via 3DPF mmand Ordering Rules)	RIMITIVE or CONSTANT_BUFFEF	 See 			
2	31	Enable								
		Format:			Enable	e				
		Specifies MI_TOPO expecting must be s	pecifies whether the HS function is enabled or disabled (pass-through). If ENABLED /II_TOPOLOGY_FILTER must be used to silently discard any topologies that the HS kernel is not xpecting. E.g., if the HS kernel is expecting PATCHLIST_32 topologies, MI_TOPOLOGY_FILTER nust be set to PATCHLIST_32 so only those topologies can reach the enabled HS.							
		Programming Notes								
		The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.								
30 Reserved										
		Format: MBZ								
	29	Statistics	Enable							
		Format:			Enable	e				
		This bit co	ontrols whet	her HS-uni	t-specific statistics registe	r(s) will increment (for each pat	ch).			
		Value	Name		Descriptio	on				
		0h	Disable	HS_INVOC	CATIONS_COUNT will not	increment				
		1h	Enable	HS_INVO	CATIONS_COUNT will incre	ement				
	28:18	Reserved				-				
		Format:				MBZ				
	17:8	Reserved								
		Format:				MBZ				
	7:4	Reserved								
		Format:				MBZ				
	3:0	Instance	Count							
		Format:	<u> </u>			<u>U4-1</u>				
		I his field If the HS	determines kernel uses	a barrier fu	r of threads (minus one) s inction software must res	pawned per input patch. trict the Instance Count to the	number			
		of threads	s that can be	e simultane	ously active within a half-	slice. Factors which must be cor	nsidered			
		includes s	cratch mem	ory availab	ility.					
		Valu	е	Name		Description				
		[0,15]			representing [1,16] instar	nces				
3	31:6	Kernel St	art Pointer							
		Format:]	nstructionE	BaseOffset[31:6]Kernel					
		This field threads s	specifies the bawned by t	e starting lo his FF unit.	ocation (1st GEN core instr It is specified as a 64-byte	ruction) of the kernel program r e-granular offset from the Instru	un by uction			



					3	DSTATE_	HS			
		Base Add	ress.							
	E-0	Reserved								
	5.0	Format:	1					MBZ		
4	31.10	Scratch S	Space	Base	Pointer					
·	51.10	Format:	pace	Dusc	Gene	ralStateOffset[31:	10]			
Specifies the location of the scratch space area allocated to this FF unit, specifie							o this FF unit, specified a	s a 1KB-		
		granular o unit will b	offset be allo	t from ocated	the Gener some po	ral State Base Add rtion of this space	lress. If req e, as specifi	uired, each thread spawr ed by Per-Thread Scratch	ned by this FF n Space.	
	9:4	Reserved								
		Format:						MBZ		
	3:0	Per-Thre	ad Sc	cratch	Space					
		Format:			U4 power	of 2 Bytes over 1	K Bytes			
	Specifies the amount of scratch space to be allocated to each thread spawned by this FF unit. The driver must allocate enough contiguous scratch space, starting at the Scratch Space Base Point to ensure that the Maximum Number of Threads can each get Per-Thread Scratch Space size without exceeding the driver-allocated scratch space								nis FF unit.The e Base Pointer, Space size	
Value Name										
		[0,11]		ir	ndicating [1K Bytes, 2M Byte	es]			
5	31:28	Reserved								
		Format:						MBZ		
	27	Single Pr Specifies with m = Execution	the in 1) or Envir	m Flo nitial as mu ronmo	ow condition ultiple prog ent.	of the kernel pro <u>c</u> gram flows (SIMD	gram as eitl Inxm with r	her a single program flov n > 1). See CR0 descripti	v (SIMDnxm on in ISA	
		Value		N	ame		Descrip	tion		
		0h	R	leserv	ed					
		1h	E	nable		Single Program F	low enable	ed		
	26	Vector M	lask E	Enabl	e					
		Format:			U1 Forma	tDesc: Enumerate	d Type			
		When SPI SPF=1, VI	F=0, \ ME sp	VME s pecifie	pecifies w s which m	hich mask to use ask to use to gen	to initialize erate execu	the initial channel enabl ution channel enables.	es. When	
		Value	Na	me		j	Descriptio	n		
		0h	Dma	ask	Channels	are enabled base	d on the di	ispatch mask		
1h Vmask Channels are enabled based on the vector mask										
	25	Reserved								
		Format:						MBZ		
	24	Include V	/erte	x Han	dles					
		Format:					Boolean			
		If set, all t	the in	put V	ertex URB	handles are inclu	ded in pay	oads. This field is ignore	d if HS	



			3D	STATE_	HS					
		Function Enable is DIS URB Entry Read Lengt	ABLED. Proc h is cleared	gramming Res to zero.	triction:Thi	s field m	nust be set if value if Verte	x		
	23:19	Dispatch GRF Start R	egister For	URB Data						
		Format:					U5			
		Specifies the starting C thread payload. This fi	GRF register ield is ignore	number for th ed if HS Functi	ne URB por on Enable	tion (Co is DISAB	onstant + Vertices) of the BLED.			
		Value			Ν	ame				
		[0,31]	indicating	GRF [R0,R31]						
	18:17	Reserved								
		Format: MBZ								
	16:11 Vertex URB Entry Read Length									
		Format:					U6			
		ayload for each Vertex UR ction Enable is DISABLED. lude Vertex Handles is clea	B ared							
		V	Name							
		[0,63]								
	10	Reserved								
		Format: MBZ								
	9:4	Vertex URB Entry Rea	ad Offset							
		Format: U6								
		Specifies the offset (in being included in the thread. This field is ign	256-bit uni thread paylo nored if HS F	ts) at which Ve bad. This offse [.] Function Enabl	ertex URB o t applies to e is DISAB	data is to all Vert LED.	b be read from the URB be tex URB entries passed to the second	fore the		
		V	/alue				Name			
		[0,63]								
	3:0	Reserved								
		Format:				MBZ				
6	31:16	Reserved								
		Format:				MBZ				
	15:13	Reserved								
		Format:				MBZ				
	12	Reserved								
		Format:				MBZ				
	11:0	Semaphore Handle								
		Format:		URBOffset[17:	:6]					
		This is the URB offset the region is 32 DWs(2 and/or HS semaphore	pointing to 16 - 512b Ul Dwords in a	the first of the RB entries). So a single contig	GS semap ftware is re juous regio	hore DV esponsib on of the	Vords in the URB. The size ole for allocating combined e URB. Software must also	of J GS		



3DSTATE_HS							
make sure the 3D pipeline is IDLE prior to allocating or deallocating the region. The semaphores can be located in an unused area within a FF unit's URB fenced region or an unused area within the Push Constant region.							



			3DS	ΓΑΤ	E_HS					
Source:		Rende	erCS							
Length B	Bias:	2								
Controls	Controls the HS stage hardware.									
DWord	Bit		Description							
0	31:29	Command	Туре							
		Default Val	ue:		3h GFXPIPE					
		Format:				ОрС	ode			
	28:27	Command	SubType							
		Default Val	ue:		3h GF	XPIPI	E_3D			
		Format:			OpCo	de				
	26:24	3D Comma	nd Opcode							
		Default Val	ue:	0h 3D	STATE_PI	PELIN	IED			
		Format:								
	23:16	16 3D Command Sub Opcode								
		Default Val	ue:		1Bh 3DSTATE_HS					
Format: OpCode										
	15:8	Reserved								
		Format:					MBZ			
	7:0	DWord Len	igth	1						
		Default Val	ue:	5 Excl	udes DW	ord ((0,1)			
		Format:		=n						
1	31:30	Reserved								
		Format:					MBZ			
	29:27	Sampler Co	ount							
		Format:	·····					U3		
		associated s	ampler state entries.	lipies o	1 4) the F	is ker	neis use	. Used only for prefetching the		
		Value	Name				Des	cription		
		0h	No Samplers	nos	samplers	used				
		1h	1-4 Samplers	bet	ween 1 ar	nd 4 s	samplers	used		
		2h	5-8 Samplers	bet	ween 5 ar	nd 8 s	samplers	used		
		3h	9-12 Samplers	bet	ween 9 ar	nd 12	sample	rs used		
		4h	13-16 Samplers	bet	ween 13 a	and 1	6 sample	ers used		
		5h-7h	Reserved							
	26	Reserved								
		Format:					MBZ			
	25:18	Binding Ta	ble Entry Count							
Format: U8								U8		



		3	BDSTATE	E_HS				
	When HW Ger Specifies how binding table e	erated Bindin many binding t ntries and asso	g Table is disat able entries the ciated surface s	oled: e kernel uses. state.	Used only for prefetching c	of the		
			Progra	mming Note	es			
	For kernels using a large number of binding table entries, it may be wise to set this field to zero to avoid prefetching too many entries and thrashing the state cache.							
17	Thread Priority Specifies the priority of the thread for dispatch Value Name							
	0h		Normal Priority	у				
	1h		High Priority					
16	Floating Point Specifies the i	Mode nitial floating	point mode u	used by the o	dispatched thread.			
	Value	Value Name Description						
	0h	IEEE-754	IEEE-754 Use IEEE-754 Rules					
	1h	1h alternate Use alternate rules						
15:14	Reserved							
	Format:				MBZ			
13	Illegal Opcode	Exception En	able					
	Format: Enable							
	This bit gets loa Execution Envir	aded into EU C onment.	R0.1[12] (note t	he bit # diffe	rence). See <i>Exceptions</i> and <i>I</i>	SA		
12	Reserved							
	Format:				MBZ			
11:8	Reserved							
	Format:				MBZ			
7	Software Exce	ption Enable						
	Format:			Enable	e			
	This bit gets los Execution Envir	aded into EU C onment.	R0.1[13] (note t	he bit # diffe	rence). See <i>Exceptions</i> and <i>I</i>	SA		
6:0	Maximum Nu	mber of Threa	ds					
	Format:		U7-1 Thread co	ount				
	Specifies the m	aximum numb	er of simultaned	ous threads a	llowed to be active. Used to	o avoid		
	using up the sc URB handles.	ratch space, or	to avoid poten	tial deadlock	. Limit is based on max num	iber of HS		
	Value		Ν	lame				
	[0,15]	indicating thre	ad count of [1,1	.6]				
			Progra	mming Note	25			



				3DSTATE_HS							
		A URB_FENCE command must be issued subsequent to any change to the value in this field and before any subsequent pipeline processing (e.g., via 3DPRIMITIVE or CONSTANT_BUFFER). See <i>Graphics Processing Engine</i> (Command Ordering Rules)									
2	31	HS Enable									
		Format: Enable									
	Specifies whether the HS function is enabled or disabled (pass-through). If ENABLED MI_TOPOLOGY_FILTER must be used to silently discard any topologies that the HS kerne expecting. E.g., if the HS kernel is expecting PATCHLIST_32 topologies, MI_TOPOLOGY_FI must be set to PATCHLIST_32 so only those topologies can reach the enabled HS.										
				Programming Note	es						
		The tessell commands otherwise	ation stages (s can only be the behavior i	(HS, TE and DS) must be enabled/ issued if all three stages are enabled is UNDEFINED.	disabled as a group. led or all three stage	I.e., draw s are disabled,					
	30	Reserved									
		Format:			MBZ						
	29	HS Statisti	cs Enable								
		Format:		Enable	5						
		This bit con	trols whether	HS-unit-specific statistics registe	r(s) will increment (fc	or each patch).					
		Value	Name	De	scription						
		0h	Disable	HS_INVOCATIONS_COUNT will n	not increment						
		1h	Enable	HS_INVOCATIONS_COUNT will in	ncrement						
	28:8	Reserved									
		Format:			MBZ						
	7:0	Instance Co	ount								
		Format:			U8						
		This field de If the HS ke of threads t includes scr	etermines the ernel uses a b hat can be sin ratch memory	e number of threads (minus one) s parrier function, software must rest multaneously active within a half-s v availability.	pawned per input pa trict the Instance Co slice. Factors which n	itch. unt to the number nust be considered					
		Value		Name							
		[0,15]	represer	nting [1,16] instances							
3	31:6	Kernel Sta	rt Pointer								
		Format:	Inst	ructionBaseOffset[31:6]Kernel							
	This field specifies the starting location (1 st GEN core instruction) of the kernel program run										
		Base Addre	ess.	The unit. It is specified as a 04-byte	-granular onset nor						
	5:0	Reserved			-						
		Format:			MBZ						
4	31:10	Scratch Sp	ace Base Poi	nter							
		Format:		GeneralStateOffset[31:10]							



				3	DS	STATE_HS				
		Specifies th granular of unit will be	ne location of fset from the allocated set of the set o	of the scr e Genera ome port	atch al Sta tion d	a space area allocated to this FF unit, specified as a 1KB- ate Base Address. If required, each thread spawned by this FF of this space, as specified by Per-Thread Scratch Space.				
	9:4	Reserved								
		Format: MBZ								
	3:0	Per-Threa	d Scratch S	pace						
		Format:	U	l power o	of 2 E	Bytes over 1K Bytes				
		Specifies the amount of scratch space to be allocated to each thread spawned by this FF unit. The driver must allocate enough contiguous scratch space, starting at the Scratch Space Base Pointer, to ensure that the Maximum Number of Threads can each get Per-Thread Scratch Spa size without exceeding the driver-allocated scratch space.								
		Value	Na	me		Description				
		[0,11]			indi	icating [1K Bytes, 2M Bytes]				
5	31:28	Reserved								
		Format:				MBZ				
	27	Specifies the initial condition of the kernel program as either a single program flow (SIMDnxm with $m = 1$) or as multiple program flows (SIMDnxm with $m > 1$). See CR0 description in <i>ISA</i> <i>Execution Environment</i> .								
		Value		Name		Description				
		0h	Reserve	ed						
		1h	Enable			Single Program Flow enabled				
	26	Vector Ma	sk Enable (VME)						
		Format:		cifies wh	Enu	umerated Type				
		SPF=1, VM	E specifies v	vhich ma	isk to	o use to generate execution channel enables.				
		Value	Name			Description				
		0h	Dmask	Channe	ls are	e enabled based on the dispatch mask				
		1h	Vmask	Channe	ls are	e enabled based on the vector mask				
	25	Reserved								
		Format:				MBZ				
	24	Include Ve	rtex Handl	es						
		Format:				Boolean				
		If set, all th	e input Vert s ignored if	ex URB ł	nandl Ttion	lles are included in payloads.				
			signored in			Programming Notes				
		This field r	nust be set	if value i	f Ver	rtex URB Entry Read Length is cleared to zero.				
	23:19	Dispatch G	GRF Start R	egister f	or Ul	JRB Data				
		Format:				U5				
		Specifies th	ne starting G	GRF regis	ter n	number for the URB portion (Constant + Vertices) of the				



			3DSTATE	HS						
		thread payload. This field is ignored if	HS Function Enable is D	ISABLED.						
		Value		Ν	lame					
		[0,31]	indicating GRF [R0,R31]							
-	18:17	Reserved	served							
		Format:			MBZ					
-	16:11	Vertex URB Entry Rea Specifies the amount of entry, in 256-bit register This field is ignored if	ad Length of URB data read and pass er increments. HS Function Enable is DIS	sed in the	thread payload <u>for each Vertex URB</u>					
		V	alue		Name					
		[0,63]								
			Due en en en							
		This field must be a p	programi	Intervention	ndles is cleared to zero					
-	10	Personned must be a m								
	10	Format [.]			MB7					
	9:4	Vertex URB Entry Rea Specifies the offset (in being included in the t thread. This field is ignored if	ad Offset 256-bit units) at which Ve hread payload. This offse HS Function Enable is DIS	ertex URB t applies t SABLED.	data is to be read from the URB before o all Vertex URB entries passed to the					
	3:0	Reserved								
		Format:			MBZ					
6 3	31:12	Reserved								
		Format:			MBZ					
	11:0	Semaphore Handle								
		Format:	U12 Ha	ndle						
	This is the 512b-aligned URB handle pointing to the first of the 32 HS semaph DWords in the URB. Software is responsible for statically allocating combined HS and/or GS semap Dwords in a single contiguous region of the URB. The semaphores can be loca unused area within a FF unit's URB fenced region or an unused area within the Constant region. This field is ignored if HS Function Enable is DISABLED.									



Source: RenderCS

Length Bias: 2

This command is used to specify the current IB state used by the VF function. At most one IB is defined and active at any given time.

NOTES: The IB must be specified before any RANDOM 3D_PRIMITIVE commands are issued It is possible to have vertex elements source completely from generated ID values and therefore not require any Index Buffer accesses. In this case, VF function will simply ignore the Index Buffer state.

DWord	Bit			Dese	riptic	n			
0	31:29	Command Type							
		Default Value:				3h GFXPIPE			
		Format:				OpCode			
	28:27	Command SubType							
		Default Value:			3h GF	XPIPE_3D			
		Format:		ОрСо	de				
	26:24	D Command Opcode							
		Default Value:		0h 3DSTA	TE_PI	PELINED			
		Format:		OpCode					
	23:16	3D Command Sub O							
		Default Value:	n 3DSTATE	INDE	X_BUFFER				
		Format:	Code						
1	15:12	Index Buffer Memor	y Object Cont	rol State					
		Format: MEMORY_OBJECT_CONTROL_STATE							
		Specifies the memory object control state for this index buffer.							
	11	Reserved							
		Format:			MBZ				
	10	Cut Index Enable							
		Format:			Enable				
		If ENABLED, the largest index value (0xFF,0xFFFF,0xFFFFFFF, depending on Index							
		Format) is interpreted as the "cut" index. (See description of this elsewhere in this section).							
		(Expected OpenGL	(Expected OpenGL driver usage). This field can only be enabled for certain primitive						
		topology types. Refer to the table later in this section for details.							
	9:8	Index Format							
		Format:	U2 enu	merated ty	pe				
		This field specifies the	e data format o	f the index	buffe	r. All index values a	are UNSIGNED.		
		Value		Nar	ne				
		0h	INDEX_BYTE						



			3DSTATE_	INDEX_BUFFER						
		1h	INDEX_WORD)						
		2h	INDEX_DWOF	RD						
	7:0	DWord Length								
		Default Value:		1h Excludes DWord (0,1)						
		Format:		=n Total Length - 2						
1	31:0	Buffer Starting A	Address							
		Format:	GraphicsAddress[3	31:0]Index_Buffer_Entry						
		the first elemer with the combi offset from the Index Buffers car	ns the size-align at of interest with nation (sum) of t base address to	he base address of the memory the starting structure within the Programming Notes in linear (not tiled) graphics memory	ry.					
2	31:0	Buffer Ending A	ddress							
		Format:	Graphics	sAddress[31:0]						
		If non-zero, this index buffer rea was zero-extend boundary (e.g.,	field contains th ds past this addr ded). Software m for an INDEX_DW	e address of the last valid byte ir ess returns an index value of 0 (a ust guarantee that the buffer end /ORD buffer, Bits [1:0] == 11b).	n the index buffer. Any as if the index buffer ds on an index					



		3DSTA	TE_LINE	_STIP	PLE					
Source:		RenderCS								
Length E	Bias:	2								
The 3DS	TATE_L	INE_STIPPLE command is used to	o specify state	variables u	used in the	e Line Stipple function.				
DWord	Bit		Description							
0	31:29	Command Type								
		Default Value:		3h GFXPIPE						
		Format:		Op	pCode					
	28:27	Command SubType								
		Default Value:		3h GFXP	IPE_3D					
		Format:		OpCode						
	26:24	3D Command Opcode								
		Default Value:	1h 3DSTATE_	NONPIPE	LINED					
		Format:	OpCode							
	23:16	3D Command Sub Opcode								
		Default Value:	08h 3DSTAT	E_LINE_ST	TIPPLE					
		Format: OpCode								
	15:8	Reserved								
		Format:								
	7:0	Dword Length								
		Default Value:	1h Excluc	les Dword	d (0,1)					
		Format:	=n Total	Length - 2	2					
1	31	Modify Enable (Current Repea	nt Counter, Cu	rrent Stip	ople Index)				
		Format: Enable								
		Modify enable for Current Repe	eat Counter an	d Curren	t Stipple I	ndex fields.				
			Program	Programming Notes						
		Software should never set this f commands as part of context sa	field to enabled ave/restore.	d. It is prov	vided only	for HW-generated				
	30	Reserved								
		Format:			MBZ					
	29:21	Current Repeat Counter								
		Format:				U9				
		This field sets the HW-internal repeat counter state. Note: Software should never attempt to this value - this state is only provided for HW-generated commands as part of context save/restore.								
	20	Reserved								
		Format:			MBZ					
	19:16	Current Stipple Index								
		Format:				U4				

г



			3DSTATE_LINE_ST	IPP	PLE				
		This field sets the HW-internal stipple pattern index. Note: Software should never attempt to set this value - this state is only provided for HW- generated commands as part of context save/restore.							
	15:0	Line Stipple Pattern							
		Format:	16 bit mask Bit 15 = most significant bit	, Bit 0 :	= least s	significant bit			
		Specifies a	pattern used to mask out bit specific pixe	els whil	le rende	ring lines.			
2	31:15	Line Stipple Inverse Repeat Count							
		Format:		U1.	16				
		Range: [0.	00390625, 1.0]						
		Specifies t	he inverse (truncated) of the repeat coun	t for th	ne line st	tipple function.			
	14:9	Reserved							
		Format:			MBZ				
	8:0	Line Stipp	le Repeat Count						
		Format: U9							
		Specifies th	ne repeat count for the line stipple function	on.					
			Value			Name			
		[1, 256]							

٦



		3DSTATE	_ M (ONOF	ILT	ER_SIZE				
Source:		RenderCS								
Length B	Bias:	2								
This state	e speci	fies the size of the filter which is	s used	when filte	ring ir	MAPFILTER_MONO mode.				
DWord	Bit			Desc	riptic	n				
0	31:29	Command Type								
		Default Value:				3h GFXPIPE				
		Format:				OpCode				
	28:27	Command SubType								
		Default Value:			3h GF	XPIPE_3D				
		Format:			ОрСо	de				
	26:24	3D Command Opcode								
		Default Value:	1h	3DSTATE_N	NONP	'IPELINED				
		Format:	Ор	Code						
	23:16	3D Command Sub Opcode								
		Default Value:	11h 3[DSTATE_M	ONOF	ILTER_SIZE				
		Format:	OpCo	de						
	15:8	Reserved								
		Format:				MBZ				
	7:0	DWord Length			- 14					
		Default Value:		0h Exclude	es DW	ord (0,1)				
		Format:		=n						
		rotal Length - 2								
1	31:6	Reserved								
		Format:				MBZ				
	5:3	Monochrome Filter Width								
		Format:				U3				
		This field specifies the width of	the m	nonochrom	e filte	r. It is ignored if the monochrome filter is				
		not enabled.			1	Nama				
		value				Name				
	2.0									
	2:0					c11				
		This field specifies the height o	f the I	monochror	ne filt	er. It is ignored if the monochrome filter is				
		not enabled.	I the i	nonocirei	ne ne	el il is ignored il the monochrome inter is				
		Value				Name				
		[1,7]								



3DSTATE_MULTISAMPLE

Source: RenderCS

Length Bias: 2

The 3DSTATE_MULTISAMPLE command is used to specify multisample state associated with the current render target/depth buffer. This is non-pipelined state.

Programming Restriction:

Driver must ierarchi that all the caches in the depth pipe are flushed before this command is parsed. This requires driver to send a PIPE_CONTROL with a CS stall along with a Depth Flush prior to this command.

When this command is issued, the currently active depth buffer, hierarchical depth buffer, stencil buffer, and render target(s) must be cleared (meaning that every pixel must be overwritten).

Alternatively, other surfaces can be activated before issuing the next 3DPRIMITIVE that were previously rendered with the same values of all state fields in this command. In other words, it is illegal to render to these surfaces with multiple different values of the state fields in this command.

Programming Notes

When programming the sample offsets (for NUMSAMPLES_4 or _8 and MSRASTMODE_xxx_PATTERN), the order of the samples 0 to 3 (or 7 for 8X) must have monotonically increasing distance from the pixel center. This is required to get the correct centroid computation in the device.

DWord	Bit		Description						
0	31:29	Command Type				-			
		Default Value:				3h GFXPIPE			
		Format:				OpCode			
	28:27	Command SubType							
		Default Value:			3h GF	XPIPE_3D			
		Format:			OpCc	ode			
	26:24	3D Command Opcode							
		Default Value:	1h 3DSTATE		PIPELINED				
		Format:		OpCode					
	23:16	3D Command Sub Opcode							
		Default Value:	0Dh 3DSTATE	E_MUL	TISAMPLE				
		Format:	OpCode						
	15:8	Reserved							
		Format:				MBZ			
	7:0	Dword Length							
		Format:	=r	n Total Length	- 2				
		Excludes Dword (0,1)							
		Value		Nar	ne				
		2h	[Defa	ault]					
1	31:6	Reserved					1		
		Format:				MBZ			



				3DSTATE_M	IULTIS	SAMPLE				
	5	Reserved	l							
		Format:				MBZ				
	4	Pixel Location								
		Format: U1								
		This field specifies where the device evaluates "pixel" (vs. centroid or sample) values/attri								
		Value		Name		Descripti	on			
		0h	PIXLO	DC_CENTER	Use the pixel center (0.5, 0.5 offset)					
		1h	PIXLO	DC_UL_CORNER	Use the pi	xel upper-left co	orner			
				F	Programm	ing Notes				
		The prog	, Iramn	ning of this field is assu	med to be	a function of the	e API being sup	ported.		
		APIs real	liy, it uire U	Is expected that OpenC L CORNER selection.	L and DXI	0+ APIS require	CEINTER Selecti	on, while DX9-		
	3:1	Number	of Mi	ultisamples						
	0.1	Format:		U3 enume	erated value	9				
		This field	l spec	cifies how many same	oles/pixel	exist in all RTs	and the Dept	า Buffer, as		
		log2(#sa	mple	s). This field is valid r	egardless	of the setting of	of Multisamp	le		
		Rasteriz	ation	Mode						
		Valu	е	Name		Descri	ption			
		0h		NUMSAMPLES_1		1 sample/pixel				
		1h		Reserved						
		2h		NUMSAMPLES_4		4 samples/pixel				
		3h		NUMSAMPLES_8		8 samples/pixel				
		[4h,7h]		Reserved						
					Programm	ing Notes				
		Setting	Mult	tisample Rasterizati	on Mode		DF XXX PATT	FRN when		
		Numbe	r of I	Multisamples == NL	JMSAMPL	ES 1 is UNDEF	INED.			
		The set	ting o	of this field must mat	ch the Nu	mber of Mult	isamples field	l in		
		SURFAC	E_ST	ATE of all bound rend	der targets	5.	-			
	0	Reserved								
		Format:				MBZ				
2	31:28	Sample3	X Of	fset						
		Format:				U0.4				
					Description	_				
		Cultural	Vaff	ent of Communic Directory		n nivel eninin Meli				
				set of Sample <u>3</u> relative S 4 or -8 Setting ignore	e to the UL ed when no	pixel origin. Vali of in MSRASTMC	d only when DF xxx PATTFI	RN		
		mode.								
		Valid whe	en NL	JMSAMPLES_1						



	Value	Name						
	[0,15]	[0,0.9375]						
27:24	Sample3 Y Offset							
	Format:	U0.4						
	Submivel V offect of Sample 2	Description	uhan					
	NUMSAMPLES_4 or _8. Settin mode.	ignored when not in MSRASTMODE_xxx_	_PATTERN					
	Valid when NUMSAMPLES_1							
	Value	Name						
	[0 15]	10 0 93751						
22.20	Sample? X Offset	[0,0.5575]						
23.20	Format:	U0.4						
	Description							
	Subpixel X offset of Sample 2	elative to the UL pixel origin. Valid only w	/hen					
	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode.	Telative to the UL pixel origin. Valid only w ignored when not in MSRASTMODE_xxx_	vhen _PATTERN					
	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1	Description relative to the UL pixel origin. Valid only w i ignored when not in MSRASTMODE_xxx_	vhen _PATTERN					
	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value	Description relative to the UL pixel origin. Valid only w gignored when not in MSRASTMODE_xxx_ Name	vhen _PATTERN					
	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15]	Description relative to the UL pixel origin. Valid only w i ignored when not in MSRASTMODE_xxx_ Name [0,0.9375]	vhen _PATTERN					
19:16	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset	Description relative to the UL pixel origin. Valid only w i ignored when not in MSRASTMODE_xxx_ Name [0,0.9375]	vhen _PATTERN					
19:16	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset Format:	Description relative to the UL pixel origin. Valid only w i ignored when not in MSRASTMODE_xxx_ Name [0,0.9375] U0.4	vhen _PATTERN					
19:16	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset Format:	Description relative to the UL pixel origin. Valid only w ignored when not in MSRASTMODE_xxx_ 0	vhen _PATTERN					
19:16	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset Format: Subpixel Y offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode.	Description relative to the UL pixel origin. Valid only w g ignored when not in MSRASTMODE_xxx_ 0	vhen _PATTERN					
19:16	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset Format: Subpixel Y offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1	Description relative to the UL pixel origin. Valid only w g ignored when not in MSRASTMODE_xxx_ 0	vhen _PATTERN					
19:16	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset Format: Subpixel Y offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1	Description relative to the UL pixel origin. Valid only w gignored when not in MSRASTMODE_xxx_ 0	vhen _PATTERN					
19:16	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset Format: Subpixel Y offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15]	Description relative to the UL pixel origin. Valid only w g ignored when not in MSRASTMODE_xxx_ [0,0.9375] U0.4 Description relative to the UL pixel origin. Valid only w ignored when not in MSRASTMODE_xxx_ Name [0,0.9375] Name [0,0.9375]	vhen _PATTERN					
19:16	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset Format: Subpixel Y offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample1 X Offset	Description relative to the UL pixel origin. Valid only w g ignored when not in MSRASTMODE_xxx_ [0,0.9375] U0.4 Description relative to the UL pixel origin. Valid only w ignored when not in MSRASTMODE_xxx_ Name [0,0.9375] Name [0,0.9375]	vhen _PATTERN					
19:16	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset Format: Subpixel Y offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample1 X Offset Format:	Description relative to the UL pixel origin. Valid only w g ignored when not in MSRASTMODE_xxx_ [0,0.9375] U0.4 Description relative to the UL pixel origin. Valid only w ignored when not in MSRASTMODE_xxx_ [0,0.9375] V0.4 Description relative to the UL pixel origin. Valid only w ignored when not in MSRASTMODE_xxx_ [0,0.9375] U0.4	vhen _PATTERN					
19:16 L5:12	Subpixel X offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample2 Y Offset Format: Subpixel Y offset of Sample <u>2</u> NUMSAMPLES_4 or _8. Settin mode. Valid when NUMSAMPLES_1 Value [0,15] Sample1 X Offset Format:	Description relative to the UL pixel origin. Valid only w g ignored when not in MSRASTMODE_xxx_ [0,0.9375] U0.4 Description relative to the UL pixel origin. Valid only w ignored when not in MSRASTMODE_xxx_ [0,0.9375] V0.4 Description relative to the UL pixel origin. Valid only w ignored when not in MSRASTMODE_xxx_ [0,0.9375] U0.4	vhen _PATTERN					



		3DSTATE_N	IULTISAMPLE							
		mode.								
		Valid when NUMSAMPLES_1								
		Value Name								
		[0,15]	[0,0.9375]							
	11:8	Sample1 Y Offset								
		Format:	U0.4							
			Description							
		Subpixel Y offset of Sample <u>1</u> relative NUMSAMPLES_4 or _8. Setting ignor mode.	e to the UL pixel origin. Valid only when red when not in MSRASTMODE_xxx_PATTERN							
		Valid when NUMSAMPLES_1								
		Value	Name							
		[0,15]	[0,0.9375]							
	7:4	Sample0 X Offset								
		Format:	U0.4							
		Description								
		NUMSAMPLES_4 or _8. Setting ignored when not in MSRASTMODE_xxx_PATTERN mode.								
		Valid when NUMSAMPLES_1								
		Value	Name							
		[0,15]	[0,0.9375]							
	3:0	Sample0 Y Offset								
		Format:	U0.4							
			Description							
		Subpixel Y offset of Sample <u>0</u> relative to the UL pixel origin. Valid only when NUMSAMPLES_4 or _8. Setting ignored when not in MSRASTMODE_xxx_PATTERN mode.								
		Valid when NUMSAMPLES_1								
		Value	Name							
			[[0,0.3575]							
3	31:28	Sample7 X Offset								
		Format:								
		Supplied X offset of Sample / relative	to the UL pixel origin. Valid only when NUMSAMPLES_8.							



	3DSTATE_M	ULTISAMPLE						
	Setting ignored when not in MSRASTMODE_xxx_PATTERN mode.							
	Value	Name						
	[0,15]	[0,0.9375]						
27:24	Sample7 Y Offset							
	Format:	U0.4						
	Subpixel Y offset of Sample <u>7</u> relative Setting ignored when not in MSRASTN	to the UL pixel origin. Valid only when NUMSAMPLES_8. MODE_xxx_PATTERN mode.						
	Value	Name						
	[0,15]	[0,0.9375]						
23:20	Sample6 X Offset							
	Format:	U0.4						
	Subpixel X offset of Sample <u>6</u> relative Setting ignored when not in MSRASTN	to the UL pixel origin. Valid only when NUMSAMPLES_8. MODE_xxx_PATTERN mode.						
	Value	Name						
	[0,15]	[0,0.9375]						
19:16	Sample6 Y Offset							
	Format:	U0.4						
	Subpixel Y offset of Sample <u>6</u> relative to the UL pixel origin. Valid only when NUMSAMPLES_8. Setting ignored when not in MSRASTMODE_xxx_PATTERN mode.							
	Value	Name						
	[0,15]	[0,0.9375]						
15:12	Sample5 X Offset							
	Format:	U0.4						
	Subpixel X offset of Sample <u>5</u> relative Setting ignored when not in MSRAST	to the UL pixel origin. Valid only when NUMSAMPLES_8. MODE_xxx_PATTERN mode.						
	Value	Name						
	[0,15]	[0,0.9375]						
11:8	Sample5 Y Offset							
	Format:	U0.4						
	Subpixel Y offset of Sample <u>5</u> relative Setting ignored when not in MSRAST	to the UL pixel origin. Valid only when NUMSAMPLES_8. MODE xxx PATTERN mode.						
	Value	Name						
	[0,15]	[0,0.9375]						
7:4	Sample4 X Offset	·						
	Format:	U0.4						
	Subpixel X offset of Sample <u>4</u> relative	to the UL pixel origin. Valid only when NUMSAMPLES_8.						
	Setting ignored when not in MSRAST	MODE_xxx_PATTERN mode.						
	Value	Name						
	[0,15]	[0,0.9375]						
3:0	Sample4 Y Offset							



3DSTATE_MULTISAMPLE								
	Format:	U0.4						
	Subpixel Y offset of Sample <u>4</u> rela	ative to the UL pixel origin. Valid only when NUMSAMPLES_8.						
	Setting ignored when not in MSR	RASTMODE_xxx_PATTERN mode.						
	Value	Name						
	[0,15]	[0,0.9375]						



		3DSTATE_PO	LY_S	STIPP	LE_	OFFSE	T	
Source:	Re	nderCS						
Length Bias:	2							
The 3DSTATE	POLY_STIP	PLE_OFFSET command is u	used to	specify th	e orig	jin of the re	peated screen-space	
Polygon Stipp	e Pattern as an X,Y offset from the Color Buffer origin.							
DWord	Bit			Desc	riptio	on		
0	31:29	Command Type						
		Default Value:				3h GFXPIP		
		Format:				OpCode		
	28:27	Command SubType			[
		Default Value:			3h G	FXPIPE_3D		
		Format:			ОрСо	ode		
	26:24	3D Command Opcode						
		Default Value:	11	n 3DSTATE	_NON	NPIPELINED		
		Format:	0	pCode				
	23:16	3D Command Sub Opco	de					
		Default Value:	06h 3D	DSTATE_PC	OLY_S	TIPPLE_OFF	SET	
		Format:	ОрСос	de				
	15:8	Reserved				L		
		Format:				MBZ		
	7:0	Dword Length					1	
		Default Value:		0h Exclue	des D	word (0,1)		
		Format:		=n Total	Leng	th - 2		
1	31:13	Reserved					1	
		Format:				MBZ		
	12:8	Polygon Stipple X Offse	t				1	
		Format:					U5	
		Specifies a 5 bit x address	offset	in the poly	y stipp I	ole pattern		
		Value					Name	
		[0,31]						
	7:5	Reserved				i		
		Format:				MBZ		
	4:0	Polygon Stipple Y Offse	t				1	
		Format:		1		-1	05	
		Specifies a 5 bit y address	onset	in the poly	y stipj	ple pattern	Nama	
		value					Name	
		[0,31]						



		3DSTATE_P	OL	Y_STIPF	PLE	_PATTERN				
Source:		RenderCS								
Length Bias: 2										
The 3DST the Polya	ATE_POL	.Y_STIPPLE_PATTERN comm le function of the WM unit.	nand is	s used to spec	ify th	ne 32x32 Polygon Stipple Pattern used in				
DWord Bit Description						on				
0	31:29	Command Type								
		Default Value:				3h GFXPIPE				
		Format:				OpCode				
	28:27	Command SubType				<u> </u>				
		Default Value:			3h G	FXPIPE_3D				
		Format:			ОрСо	ode				
	26:24	3D Command Opcode	3D Command Opcode							
		Default Value:	: 1h 3DSTA		NON	PIPELINED				
		Format: OpCode								
	23:16	3D Command Sub Opcoc	de							
		Default Value:	07h 3	h 3DSTATE_POLY_STIPPLE_PATTERN						
		Format:	ОрСо	de						
	15:8	Reserved								
		Format: MBZ								
	7:0	Dword Length								
		Default Value:		1Fh Excluc	1Fh Excludes Dword (0,1)					
		Format:		=n Total L	ength	n - 2				
1	31:0	Polygon Stipple Pattern	Row 1	. (top most)		1				
		Format: 32 bit mask Bit 3	31 = u	pper left corr	ner, Bi	it 0 = upper right corner of first row.				
		Specifies a pattern used by Polygon Stipple to mask out specific pixels of every 32x32 area rendered.								
232	31:0	Polygon Stipple Pattern	Rows	2-32 (bottor	n mo	st)				
		Format: 32 bit mask Bit 3	31 = u	pper left corr	ner, Bi	it 0 = upper right corner of first row.				
		Specifies a pattern used by Polygon Stipple to mask out specific pixels of every 32x32 area rendered.								



			30	DS.	ΓΑΤΕ	_PS					
Source:		Render	CS								
Length Bias: 2											
DWord	Bit		Description								
0	31:29	Command Ty	pe								
		Default Value	:				3h G	FXPIPE			
		Format:					ОрС	ode			
	28:27	Command Su	ьТуре								
		Default Value	:			3h GF	XPIPI	E_3D			
		Format:				OpCo	de				
	26:24	3D Command	l Opcode								
		Default Value	0h 3DST	ATE_PI	PELIN	IED					
		Format:			OpCode	è					
	23:16	3D Command	l Sub Opcode								
		Default Value	:		:	20h 3DS	STATE	E_PS			
		Format:			(OpCode	ġ				
	15:8	Reserved									
		Format:	mat: MBZ								
	7:0	DWord Length									
		Default Value	06h Excludes DWord (0,1)								
		Format: =n									
		Total Length -	2								
1	31:6	Kernel Start Pointer[0]									
		Format:	InstructionB	laseC	Offset[31:	6]Kerne	el				
		Specifies the 64-byte aligned address offset of the first instruction in the kernel[0]. This pointer is relative to the Instruction Base Address.)]. This pointer is		
	5:0	Reserved									
		Format:						MBZ			
2	31	Single Progra Specifies the i with m = 1) or Execution Envi Value	m Flow (SPF) nitial condition of as multiple progra ronment.	the k am fl	kernel pr ows (SIN	ogram a IDnxm v De	as eitl with r scrip	her a single program fl m > 1). See CR0 descri	low (SIMDnxm ption in ISA		
		Oh	Multiple	Mul	tiple Pro	aram El					
		1h	Single	Sino	ile Proar	am Flow	//s				
	20	Voctor Mask		Jing	, e riogi						
	30	Format		- num	nerated 1	vne					
		When $SPF=0$		h m	ask to us	e to init	ialize	the initial channel ena	ables When		
		When SPF=0, VME specifies which mask to use to initialize the initial channel enables. When SPF=1. VME specifies which mask to use to generate execution channel enables.									



			30	STATE_F	PS					
	Value	Name		De	scription					
	0h	Dmask	Channels are	enabled based	on the dispatch	mask				
	1h	Vmask Channels are enabled based on the vector mask						1		
29:27	Sampler	Sampler Count								
	Format: U3									
	Specifies prefetchin	how many	samplers (in	multiples of 4) tł er state entries.	ne pixel shader 0	kernel uses. l	Jsed o	nly for		
	Value	Nan	ne	D	escription					
	[0,4]									
	0h		no sam	olers used						
	1h		betweer	n 1 and 4 sample	ers used					
	2h		betweer	n 5 and 8 sample	ers used					
	3h		betweer	n 9 and 12 samp	lers used					
	4h		betweer	n 13 and 16 sam	plers used					
	5h-7h		Reserve	d						
26	Denormal Mode									
	Specifies	the denor		Description						
	Oh	FT7	Denor	enormals are flushed to zero						
	1h	RFT	Denor	mals are retaine	als are retained					
25.18	Binding '	Table Enti	The Count							
23.10	Format:									
	Specifies binding ta binding ta many ent This field	how many able entrie able entrie ries and th l is ignored	binding table s and associates, it may be a arashing the st d if [PS Function	e entries the kerr red surface state dvantageous to s rate cache. on Enable] is DIS.	nel uses. Used or . Note: For kerne set this field to z ABLED.	ly for prefetcl els using a larg ero to avoid p	hing of ge num prefetch	the ber of ning too		
			Value			Name				
	[0,255]							_		
	Programming Notes									
	When H generate	W binding ed at JIT tir	table bit is se ne.	t, it is assumed t	hat the Binding ⁻	Table Entry Co	ount fie	ld will be		
17	Reserved									
	Format:				MBZ					
16	Floating Specifies	Point Mo the floatin	de g point mode	used by the dis	oatched thread.					
	Valu	e	Name		Description					
	0h	IEE	E-745	Use IEEE-754	4 rules					
	1h	Alt		Use alternat	e rules					



				3DSTATE PS						
	15:14	Rounding Mode								
		Specifies the r								
		Value	Name	Description Description	ion					
		Un	RINE	Round to Nearest Even						
		In								
		2h	RD	Round toward -infinity						
		3h	RTZ	Round toward zero						
	13	Illegal Opcod	e Exception E	nable						
		Format:		Enable	e					
		This bit gets lo Execution Envi	baded into EU (ironment.	CR0.1[12] (note the bit # diffe	rence). See Exception:	s and ISA				
	12	Reserved								
		Format:			MBZ					
	11	Mask Stack E	xception Enab	le	1					
		Format:	•	Enable	e					
		This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.								
	10:8	Reserved								
		Format:			MBZ					
	7	Software Exc	eption Enable		•					
		Format: Enable								
		This bit gets loaded into EU CR0.1[13] (note the bit # difference). See Exceptions and ISA Execution Environment.								
	6:0	Reserved								
		Format:			MBZ					
3	31:10	Scratch Space	e Base Pointer							
		Format:	GeneralSta	ateOffset[31:10]ScratchSpace						
		Specifies the 1k-byte aligned address offset to scratch space for use by the kernel. This pointer is relative to the General State Base Address .								
	9:4	Reserved								
		Format:			MBZ					
	3:0	Per Thread So	cratch Space		•					
		Format:	•		U4					
		Specifies the amount of scratch space allowed to be used by each thread. The driver must allocate enough contiguous scratch space, pointed to by the Scratch Space Pointer, to ensure that the Maximum Number of Threads each get Per Thread Scratch Space size without exceeding the driver-allocated scratch space.								



				3DST/	ATE_PS	5					
		Value				Name					
		[0,11]	indicati	ndicating [1k bytes, 2M bytes] in powers of two							
4	31:24	Maximum Number of Threads									
		Format: U8-1 representing thread count									
		Description									
		Description									
		WIZ Hashing Disable in GT_MODE register enabled: Range = [7,171]> [8,172] threads. Only odd values are allowed (resulting in even max number of threads) WIZ Hashing Disable in GT_MODE register disabled: Range = [3,85]> [4,86] threads. Only odd values are allowed (resulting in even max number of threads)									
		Specifies the avoid using	e maxim up the s	um number of simu cratch space, or to	ultaneous th avoid poter	reads allowed to be actinitial deadlock.	ve. Used to				
		Valu	e	Name		Description]				
		[3h,1fh]	-	Range	[4,32] threa	ads	-				
							J				
		Programming Notes									
		If this field is changed between 3DPRIMITIVE commands, a PIPE_CONTROL command with Stall at Pixel Scoreboard set is required to be issued. This field must have an odd value so that the max number of PS threads is even.									
	23:12	Reserved									
		Format:				MBZ					
	11	Push Constant Enable									
		Format:				Enable					
		This field must be enabled if the sum of the PS Constant Buffer [3:0] Read Length fields in 3DSTATE_CONSTANT_PS is nonzero, and must be disabled if the sum is zero.									
	10	Attribute En	able								
		Format:				Enable					
		This field must be enabled if the Number of SF Output Attributes field in 3DSTATE_SBE is nonzero, and must be disabled if that field is zero.									
	9	oMask Pres	ent to R	enderTarget							
		Format:				Enable					
		This bit is ins message hea included in R samples.	erted in Ider or v Render T	the PS payload hea ia header bypass) to arget Write messag	ader and ma o indicate th Jes. If preser	ade available to the Data nat oMask data (one or t nt, the oMask data is use	Port (either via the wo phases) is d to mask off				
	8	Render Targ	jet Fast	Clear Enable							
		Format:				Enable					



		31	DSTATE_PS						
	This field is set to enable fast clear of the bound render targets. See "Render Target Fast Cle for restrictions on enabling this field.								
7	Dual Source Blend Enable								
	Forma	it:	Enable						
	This field is set if dual source blend is enabled. If this bit is disabled, the data port dual source message reverts to a single source message using source 0.								
6	Rende	r Target Resolve Enable							
	Forma	at:	Enable						
	This fie Target	eld is set to enable clear v Resolve" for restrictions o	alue resolve on non-multisampled render targets. See "Re on enabling this field.	ender					
5	Reserv	ved							
	Forma	at:	MBZ						
4:3	Positio	on XY Offset Select							
	Forma	at: U2	Enumerated Type						
	This field specifies if/what Position XY Offset values are passed in the PS payload. Note that these are per-slot (pixel sample) offsets, and therefore separate from the subspan XY coordinates passed in R1.								
	Value	Name	Description						
	0h	POSOFFSET_NONE	No Position XY Offsets are included in the PS payload.						
	1h	Reserved							
	2h	POSOFFSET_CENTROID	Position XY Offsets will be passed in the PS payload, and these will reflect the Centroid position(s).						
	3h	POSOFFSET_SAMPLE	Position XY Offsets will be passed in the PS payload, and these will reflect the multisample position(s).						
			Programming Notes						
	SW Re value, positio	SW Recommendation: If the PS kernel needs the Position Offsets to compute a Position XY value, this field should match Position ZW Interpolation Mode to ensure a consistent position.xyzw computation							
	If the field s variou	If the PS kernel does not need the Position XY Offsets to compute a Position Value, then this field should be programmed to POSOFFSET_NONE, as the PS kernel should be using the various barycentric inputs to evaluate other-than-position attributes.							
	MSDIS	MSDISPMODE_PERSAMPLE is required in order to select POSOFFSET_SAMPLE.							
2	32 Pix	el Dispatch Enable							
	Forma	it:	Enable						
			Description						
	[reak!	a tha Windower to diara	tsh 9 subspans in one poulos d						
		es the windower to dispa	low the Valid column indicates which products that						
	inote:	See Note. In the table be	now, the value column mulcates which products that						



	3DSTATE_PS					
	 combination is supported on. Combinations of dispatch enables not listed in the table are not available on any product. A: Valid on all products B: Valid. C: Not valid. D: Valid on all products, except when in non-1x PERSAMPLE mode. E: Valid on all products, except when in PERSAMPLE mode with number of multisamples > = 8. F: Valid on all products. 					
	Each of the three KSP values are separately specified.					
	In addition, each kernel has a separately-specified GRF register count.					
	Variable Pixel Dispatch Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.					
1	16 Pixel Dispatch Enable					
	Format: Enable					
	Description					
	Enables the Windower to dispatch 4 subspans in one payload.					
	Note: See Note: in the table below, the Valid column indicates which products that combination is supported on. Combinations of dispatch enables not listed in the table are not available on any product. A: Valid on all products B: Valid. C: Not valid. D: Valid on all products, except when in non-1x PERSAMPLE mode. E: Valid on all products, except when in PERSAMPLE mode with number of multisamples > = 8. F: Valid on all products.					
	Each of the three KSP values are separately specified.					
	In addition, each kernel has a separately-specified GRF register count.					
	Variable Pixel Dispatch Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.					
0	8 Pixel Dispatch Enable					
	Format: Enable					
	Description					
	Enables the Windower to dispatch 2 subspans in one payload. Note: See Note: in the table below, the Valid column indicates which products that combination is supported on. Combinations of dispatch enables not listed in the table are not available on any product. A: Valid on all products B: Valid					



3DSTATE_PS				
	C: Not valid.D: Valid on all products, except when in non-1x PERSAMPLE mode.E: Valid on all products, except when in PERSAMPLE mode with number of multisamples > = 8.F: Valid on all products.Each of the three KSP values are separately specified.In addition, each kernel has a separately-specified GRF register count.Variable Pixel Dispatch Section: Pixel Grouping (Dispatch size) control for valid pixel			
-	dispatch combinations.			
5	31:23	Format:	MBZ	
	22.16	Dispatch GRF Start Register for Constant/Setup	p Data [0]	
		Format:	U7	
		Specifies the starting GRF register number for the Constant/Setup portion of the thread payload for kernel[0].		
		Value	Name	
		[0,127]		
	15	Reserved	1	
		Format:	MBZ	
	14:8	Dispatch GRF Start Register for Constant/Setup Data [1]		
		Format:	U/	
		for kernel[1].	constant/setup portion of the thread payload	
		Value	Name	
		[0,127]		
	7	Reserved		
		Format:	MBZ	
	6:0	Dispatch GRF Start Register for Constant/Setup Data [2]		
		Format:	U7	
		Specifies the starting GRF register number for the Constant/Setup portion of the thread payload for kernel[2].		
		Value	Name	
		[0,127]		
6	31:6	Kernel Start Pointer[1]		
		Format: InstructionBaseOffset[31:6]Kernel		
		relative to the Instruction Base Address.	e first instruction in kernel[1]. This pointer is	
	5:0	Reserved		
		Format:	MBZ	


	3DSTATE_PS									
7	31:6	Kernel Start F	Kernel Start Pointer[2]							
		Format:	Format: InstructionBaseOffset[31:6]Kernel							
		Specifies the 64-byte aligned address offset of the first instruction in kernel[2]. This pointer is relative to the Instruction Base Address .								
	5:0	Reserved								
		Format: MBZ								



	3DSTATE_PUSH_CONSTANT_ALLOC_DS										
Source:		RenderCS									
Length B	ias:	2									
This com	mand	sets up the URB configura	ation fo	r DS	Push Cons	tant B	uffer.				
			Pr	ogra	amming N	otes					
Program	nming F	Restriction:									
• T 0	 The sum of the Constant Buffer Offset and the Constant Buffer Size may not exceed the maximum value of the Constant Buffer Size. 										
• T tl	 The sum of the constant length programmed in 3DSTATE_CONSTANT_DS must be equal or smaller then the size of the allocated space in the URB including the buffering for half cachelines. See Push Constant URB Allocation section for more details. 										
• T	he 3DS rogram	TATE_CONSTANT_DS mu nming the 3DSTATE_PUSI	ist be re H_CONS	eprog TAN	grammed p IT_ALLOC_	orior to DS.	o the	next 3D	PRIMITIVE command after		
DWord	Bit	Description									
0	0 31:29 Command Type										
		Default Value:					3h G	FXPIPE			
		Format: OpCode									
	28:27	Command SubType									
		Default Value:				3h GF	XPIPE	_3D			
		Format:				ОрСо	de				
	26:24	3D Command Opcode		1							
				In	3DSTATE_						
		Format:		Op	Code						
	23:16	3D Command Sub Opcode									
			14n 3L	51A	TE_PUSH_C	USH_CONSTANT_ALLOC_DS					
	17.0	Format:	Ορυος	ie							
	15:8	Reserved						MDZ			
	7.0	Format.						IVIDZ			
	7:0	Dword Length			Oh Evolud		ord ((11)			
		Format:				onath		J, L)			
1											
T	31:20	Format:						MB7			
	10.16	Constant Puffer Offeet									
	19:10	Format:							114		
		Specifies the offset of th	e DS co	nsta	nt buffer ir	to the	URB				
		Value			,	,		Nam	e		
[0,15] (0KB - 15KB)											



3DSTATE_PUSH_CONSTANT_ALLOC_DS									
	0h		0KB [Default]						
15:5	Reserved								
	Format:	at: MBZ							
4:0	Constant Buffer Size								
	Format:				U5				
	Specifies the size of command stream constants are not	of the DS constant buffer. This value will determine the amount of data the trican pre-fetch before the buffer is full. Value of zero is only valid when t enabled for DS							
	Value		Nar	ne					
	[0,15]	(OKB - 15KB) Increments of 1KB							
	0h	0KB [Defaul t	t]						



3DSTATE_PUSH_CONSTANT_ALLOC_GS

Source:

RenderCS Length Bias: 2

This command sets up the URB configuration for GS Push Constant Buffer.

Programming Notes

- The sum of the Constant Buffer Offset and the Constant Buffer Size may not exceed the maximum value of the Constant Buffer Size.
- The sum of the constant length programmed in 3DSTATE CONSTANT GS must be equal or smaller then the size of the allocated space in the URB including the buffering for half cachelines.
- The 3DSTATE_CONSTANT_GS must be reprogrammed prior to the next 3DPRIMITIVE command after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_GS.

See Push Constant URB Allocation section for more details.

DWord	Bit				Dese	riptio	n			
0	31:29	Commar	nd Type							
		Default '	Value:				3h G	FXPIPE		
		Format:					OpCode			
	28:27	Commar	nd SubType							
		Default '	Value:			3h GF	3h GFXPIPE_3D			
		Format:		ОрС						
	26:24	3D Com	mand Opcode							
		Default '	Value:		1h 3DSTATE_	NONPI	[PELI]	NED		
		Format:			OpCode					
	23:16	3D Com	D Command Sub Opcode							
		Default '	Value:	15h 3D	STATE_PUSH_(CONST	ANT_	ALLOC_	GS	
		Format:		OpCod	e					
	15:8	Reserved	ł							
		Format:						MBZ		
	7:0	DWord I	_ength							
		Format:							=n	
		Total Len	gth - 2							
		Value			Name					Description
		0h	3DSTATE_PUSH	I_CONS	TANT_ALLOC_(GS [De	fault]		Excludes DWord (0,1)
1	31:20	Reserved	1							1
		Format:						MBZ		
	19:16	Constan	t Buffer Offset							
Format: U4										
		Specifies	the offset of the	e GS cor	nstant buffer in	to the	URB.			
			Value					Nam	е	



3DSTATE_PUSH_CONSTANT_ALLOC_GS									
	[0,15]		(0KB - 15KB)						
	0h		OKB [Default]						
15:5	Reserved								
	Format:	MBZ							
4:0	Constant Buffer Size								
	Format:				U5				
	Specifies the size of command stream constants are not	of the GS con: can pre-fetch enabled for G	stant buffer. This value will before the buffer is full. Va iS.	determi alue of z	ne the amount of data the ero is only valid when				
	Value		Nar	ne					
	[0,15]	(OKB - 15KB)	(OKB - 15KB) Increments of 1KB						
	0h	0KB [Defaul t	t]						



	3DSTATE_PUSH_CONSTANT_ALLOC_HS										
Source:		RenderCS									
Length B	ias:	2									
This com	mand s	sets up the URB configur	ation fo	r HS	Push Cons	stant B	uffer.	,			
			Pr	ogra	amming N	otes			T		
Program	nming F	Restriction:									
 The sum of the Constant Buffer Offset ar of the Constant Buffer Size. The sum of the constant length program the size of the allocated space in the UR URB Allocation section for more detail The 3DSTATE_CONSTANT_HS must be re programming the 3DSTATE_PUSH_CONSTANT_ 					and the Constant Buffer Size may not exceed the maximum value ammed in 3DSTATE_CONSTANT_HS must be equal or smaller then JRB including the buffering for half cachelines. See Push Constant tails. e reprogrammed prior to the next 3DPRIMITIVE command after NSTANT_ALLOC_HS.						
DWord Bit Description											
0	31:29	Command Type				÷					
		Default Value:	fault Value:				3h GFXPIPE				
Format:				OpCode				ode			
	28:27	Command SubType									
		Default Value:				3h GF	XPIPE	E_3D			
		Format:		OpCode							
	26:24	3D Command Opcode									
		Default Value:	efault Value: 1			1h 3DSTATE_NONPIPELINED					
		Format:	Op	OpCode							
	23:16	3D Command Sub Opcode									
		Default Value:	13h 3[DSTA	TE_PUSH_CONSTANT_ALLOC_HS						
		Format:	OpCod	de							
	15:8	Reserved							_		
		Format:						MBZ			
	7:0	DWord Length									
		Default Value:			0h Exclud	es DW	ord ((0,1)	_		
		Format:			=n Total L	.ength	- 2				
1	31:20	Reserved							_		
		Format:			MBZ						
	19:16	Constant Buffer Offset							_		
		Format:	110				1100	U4			
		Specifies the offset of th	ie HS co	onsta	nt buffer in	nto the	URB	Namo	٦		
		10 151			2 - 15VP)			IVAILLE			
[[0,15] (0KB - 15KB)											



3DSTATE_PUSH_CONSTANT_ALLOC_HS									
	0h	0h 0KB [Default]							
15:5	Reserved								
	Format:	ormat: MBZ							
4:0	Constant Buffer Size								
	Format:				U5				
	Specifies the size command stream constants are not	of the HS con can pre-fetch enabled for H	stant buffer. This value will before the buffer is full. Va IS.	determi alue of z	ine the amount of data the ero is only valid when				
	Value		Nar	ne					
	[0,15] (OKB - 15KB) Increments of 1KB								
	0h	0KB [Defaul t	t]						



3DSTATE_PUSH_CONSTANT_ALLOC_PS

Source:

RenderCS

Length Bias: 2

Description

This command sets up the URB configuration for PS Push Constant Buffer.

A PIPE_CONTOL command with the CS Stall bit set must be programmed in the ring after this instruction.

Programming Notes

Restriction:

- The sum of the Constant Buffer Offset and the Constant Buffer Size may not exceed the maximum value of the Constant Buffer Size.
- The sum of the constant length programmed in 3DSTATE_CONSTANT_PS must be equal or smaller then the size of the allocated space in the URB including the buffering for half cachelines. See **Push Constant URB Allocation** section for more details.
- The 3DSTATE_CONSTANT_PS must be reprogrammed prior to the next 3DPRIMITIVE command after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_PS.

DWord	Bit		Description							
0	31:29	Command Type								
		Default Value:				3h GFXPIPE				
		Format:				OpCode				
	28:27	Command SubType								
		Default Value:		3h G	FXPIPE	E_3D				
		Format:		ОрСо	ode					
	26:24 3D Command Opcode									
		Default Value:	1h 3DSTA	FE_NON	PIPELIN	NED				
		Format:	OpCode							
	23:16	3D Command Sub Opco	ode							
		Default Value:	ult Value: 16h 3DSTATE_PUSH			TANT_	_ALLOC_PS			
		Format:	OpCode	2						
	15:8	Reserved								
		Format:			MBZ					
	7:0	Dword Length								
		Default Value:		0h Exc	0h Excludes Dword (0,1)					
		Format:		=n To	=n Total Length - 2					
1	31:20	Reserved								
		Format:				MBZ				
	19:16	Constant Buffer Offset								
		Format:					U4			



3DSTATE_PUSH_CONSTANT_ALLOC_PS								
	Specifies the offse	t of the PS co	nstant buffer into the URB	•				
	Value	e	Name					
	[0,15]		(0KB - 15KB)					
	0h		OKB [Default]					
15:5	Reserved	Reserved						
	Format:		MBZ					
4:0	Constant Buffer	Size						
	Format:				U5			
	Specifies the size of command stream constants are not	cifies the size of the PS constant buffer. This value will determine the amount of data the mand stream can pre-fetch before the buffer is full. Value of zero is only valid when stants are not enabled for PS.						
	Value	Name						
	[0,15]	(OKB - 15KB) Increments of 1KB						
	0h	OKB [Defaul	t]					



Source:

RenderCS Length Bias: 2

This command sets up the URB configuration for VS Push Constant Buffer.

Programming Notes

Programming Restriction:

- The sum of the Constant Buffer Offset and the Constant Buffer Size may not exceed the maximum value of the Constant Buffer Size.
- The sum of the constant length programmed in 3DSTATE_CONSTANT_VS must be equal or smaller then the size of the allocated space in the URB including the buffering for half cachelines. See **Push Constant** URB Allocation section for more details.
- The 3DSTATE_CONSTANT_VS must be reprogrammed prior to the next 3DPRIMITIVE command after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_VS.

DWord	Bit			Dese	criptio	n				
0	31:29	Command Type								
		Default Value:				3h G	GFXPIPE			
		Format:				OpCode				
	28:27	Command SubType								
		Default Value:	ult Value: 3r				PE_3D			
		Format:			ОрСо	de				
	26:24	3D Command Opcode								
		Default Value: 1h 3D			NONP	IPELI	INED			
		Format:	OpCode							
	23:16	D Command Sub Opcode								
		Default Value: 12h 3DSTATE_PUSF				TANT	r_alloc_vs			
		Format:	OpCod	e						
	15:8	eserved								
		Format:			MBZ					
	7:0	DWord Length								
		Default Value:		0h Exclud	0h Excludes DWord (0,1)					
		Format:		=n Total L	Length - 2					
1	31:20	Reserved								
		Format:					MBZ			
	19:16	Constant Buffer Offset								
		Format:				U4				
		Specifies the offset of the	e VS cor	istant buffer ir	nto the	URB	3.			
		Value			Name					
		[0,15]	(OKB - 15KB)						



	3DSTATE_PUSH_CONSTANT_ALLOC_VS									
		0h		0KB [Default]						
1	15:5	Reserved	leserved							
		Format:	Format: MBZ							
	4:0	Constant Buffer Size								
		Format:				U5				
		Specifies the size of command stream constants are not	of the VS cons can pre-fetch enabled for V	stant buffer. This value will before the buffer is full. Va 'S.	determi alue of z	ne the amount of data the ero is only valid when				
		Value		Nar	ne					
		[0,15] (0KB - 15KB) Increments of 1KB								
		0h	OKB [Defaul t	t]						



		3DSTATI	E_S	SAMP	LE_	M	ASK		
Source:		RenderCS							
Length B	Bias:	2							
DWord	Bit			Desc	criptio	n			
0	31:29	Command Type							
		Default Value:				3h G	FXPIPE		
		Format:				ОрС	ode		
	28:27	Command SubType	/pe						
		Default Value:			3h GFX	XPIPE	E_3D		
		Format:			OpCod	de			
	26:24	3D Command Opcode							
		Default Value:		0h 3DSTA	ATE_PIP	'ELIN	IED		
		Format:	OpCode	JpCode					
	23:16	3D Command Sub Opcode							
		Default Value:	18h	3DSTATE_	SAMPL	LE_M	IASK		
		Format:	OpC	Lode					
	15:8	Reserved							
		Format:					MBZ		
	7:0	Dword Length							
		Default Value:		0h Exclud	les Dword (0,1)				
		Format:		=n Total I	tal Length - 2				
1	31:8	Reserved							
		Format:					MBZ		
	7:0	Sample Mask							
		Format: 8 bit mask Right-justifie determined by Num M	ed bi [.] ultisa	itmask (Bit amples (3[: 0 = Sa DSTATE	ample E_MU	e0). Number of bits that are used is JLTISAMPLE)		
		A per-multisample-position mask with the sample coverage mask a	k stat as pa	te variable art of the r	e that is asteriza	; imn ation	nediately and unconditionally ANDed process. This mask is applied prior to		
		centroid selection.					-		
				Program	ming I	Note	!5		
		If Number of Multisamp	oles i	is NUMSA	MPLES	_1, bi	its 7:1 of this field must be zero.		
		• If Number of Multisamp	oles i	is NUMSA	MPLES	_4, bi	its 7:4 of this field must be zero.		



		3DSTATE_SAI	MPLE	R_PAL	ET	TE_LOAD0			
Source:		RenderCS							
Length Bias:		2							
		De	escription						
The 3DSTA texture pale "Px [palette	TE_SAMPL ette. The t 0]") is refe	ER_PALETTE_LOAD0 inst exture palette is used wh erenced by the sampler.	ruction is unenever a t	used to load exture with	d 32 a pa	-bit values into the first aletted format (containing			
This instruct always start	tion is use from the	ed to load all or a subset first (index 0) entry.	of the 256	entries of t	he f	first palette. Partial loads			
DWord	Bit			Descr	ipti	on			
0	31:29	Command Type				1			
		Default Value:				3h GFXPIPE			
		Format:				Opcode			
	28:27	Command SubType							
		Default Value:		3	3h G	FXPIPE_3D			
		Format:		C	Эрсо	ode			
	26:24	3D Command Opcode	•						
		Default Value:				1h 3DSTATE			
		Format:				Opcode			
	23:16	3D Command Sub Op	code						
		Default Value:	02h 3DS	TATE_SAMP	LER	_PALETTE_LOAD0			
		Format: Opcode							
	15:8	Reserved							
		Format:	Format:				MBZ		
	7:0	DWord Length							
		Default Value:		0h Exclude	es D\	Word (0,1)			
		Format:		=n					
		Total Length - 2	Total Length - 2						
1n	31:24	Palette Alpha[0:N-1]							
		Format:				U8			
		Alpha channel loaded into the Nth entry of the texture color palette.							
	23:16	Palette Red[0:N-1]							
		Format:				U8			
		Alpha channel loaded in	nto the Nt	h entry of th	he te	exture color palette.			
	15:8	Palette Green[0:N-1]							
		Format:				U8			
		Alpha channel loaded in	nto the Nt	h entry of th	he te	exture color palette.			



3DSTATE_SAMPLER_PALETTE_LOAD0								
7:0	Palette Blue[0:N-1]							
	Format:	U8						
	Alpha channel loaded into the Nth entry of the texture color	oalette.						



		3DSTATE_S	AMPL	.ER_P/	4L	.ETTE_LOA	D1		
Source:		RenderCS							
Length Bia	as:	2							
The 3DST/ palette. Th "Px[palet the second	ATE_SAN ie secon tte1]") is d palette	IPLER_PALETTE_LOAD1 in Id texture palette is used referenced by the sample e. Partial loads always sta	nstruction i whenever a er. This ins rt from the	is used to le a texture w truction is first (index	oad 'ith use (0)	32-bit values into a paletted format (d to load all or a su entry.	the second texture containing bset of the 256 entries of		
DWord	Bit			Desc	Description				
0	31:29	Command Type					I		
		Default Value:				3h GFXPIPE			
		Format:				OpCode			
	28:27	28:27 Command SubType							
		Default Value:			3h GFXPIPE_3D				
		Format:		Code					
	26:24	3D Command Opcode							
		Default Value:				1h 3DSTATE			
		Format:				OpCode			
	23:16	3D Command Sub Opc	ode						
		Default Value:	0Ch 3DST	rate_samf	PLEF	R_PALETTE_LOAD1			
		Format:	OpCode						
	15:8	Reserved							
		Format:				MBZ			
	7:0	DWord Length					1		
		Default Value:		0h Exclud	es [DWord (0,1)			
		Format:		=n Total I	_eng	gth - 2			
1n	31:24	Palette Alpha[0:N-1]							
		Format:				U8			
		Alpha channel loaded in	to the Nth	entry of th	e te	exture color palette.			
	23:16	Palette Red[0:N-1]							
		Format:				U8			
		Alpha channel loaded in	Alpha channel loaded into the Nth entry of the texture color palette.						
	15:8	Palette Green[0:N-1]					1		
		Format:				U8			
		Alpha channel loaded in	to the Nth	entry of th	e te	exture color palette.			
	7:0	Palette Blue[0:N-1]							
		Format:				U8			
		Alpha channel loaded in	to the Nth	entry of th	e te	exture color palette			



	3	BDSTATE_SA	MPLER	_STA	TE_	POINTERS_DS	
Source:		RenderCS					
Length Bia	as:	2					
The 3DST table. Only	ATE_SA	MPLER_STATE_POINTER	RS_DS comma tilize sampler	ind is used state tabl	d to de es.	lefine the location of DS SAMPLER_STATE	
DWord	Bit			Des	criptio	on	
0	31:29	Command Type					
		Default Value:				3h GFXPIPE	
		Format:				OpCode	
	28:27 Command SubType						
		Default Value:			3h Gl	FXPIPE_3D	
		Format:			OpCode		
	26:24	3D Command Opcod	e				
		Default Value:	Default Value:		ATE_P	PIPELINED	
		Format:		OpCode			
	23:16	3D Command Sub Op	ocode				
		Default Value:	2Dh 3DSTA1	DSTATE_SAMPLER_STATE_POINTERS_DS			
		Format:	OpCode				
	15:8	Reserved					
		Format:				MBZ	
	7:0	DWord Length					
		Default Value:		0h DW0	0h DWORD_COUNT_n		
		Format:		=n			
1	31:5	Pointer to DS Sample	er State				
		Format: Dynam	nicStateOffset	[31:5]SAN	1PLER	R_STATE*16	
		Specifies the 32-byte offset is relative to the	aligned addre Dynamic Stat	ess offset o te Base Ac	of the ddress	DS function's SAMPLER_STATE table. This s.	
	4:0	Reserved					
		Format:				MBZ	



		3DSTATE_SA	MPLER	R_STA	TE_P	POINTERS_GS		
Source:		RenderCS						
Length B	ias:	2						
The 3DS table. Or	TATE_S	AMPLER_STATE_POINTE	RS_GS comma utilize sampler	and is use state tab	ed to defi les.	ine the location of GS SAMPLER_STATE		
DWord	Bit			Dese	cription			
0	31:29	Command Type						
		Default Value:			3h	h GFXPIPE		
		Format:	ormat:			pCode		
	28:27	Command SubType						
		Default Value:			3h GFXP	PIPE_3D		
		Format:			OpCode	2		
	26:24	3D Command Opcode						
		Default Value:		0h 3DSTA	ATE_PIPEL	LINED		
		Format:	OpCode					
	23:16	3D Command Sub Ope	code					
		Default Value:	2Eh 3DSTATE_SAMPLER_STATE_POINTERS_GS					
		Format:						
	15:8	Reserved						
		Format:		MBZ				
	7:0	DWord Length						
		Default Value:		0h DWORD_COUNT_n				
		Format:		=n				
1	31:5	Pointer to GS Sampler	State					
		Format: DynamicStateOffset[31:5]SAMPLER_STATE*16						
		Specifies the 32-byte a offset is relative to the I	ligned addres: Dynamic State	s offset of Base Ado	f the GS f dress.	function's SAMPLER_STATE table. This		
	4:0	Reserved						
		Format:				MBZ		

г



		3DSTATE_S	AMPLER	R_STA	TE_P	OINTERS_I	HS		
Source:		RenderCS							
Length B	ias:	2							
The 3DS table. Or	TATE_S	AMPLER_STATE_POINT	ERS_HS comm utilize sample	and is use r state tab	ed to defi	ne the location of H	S SAMPLER_STATE		
DWord	Bit			Des	cription				
0	31:29	Command Type							
		Default Value:	ault Value:			3h GFXPIPE			
		Format:			OpCode				
	28:27	Command SubType	Command SubType						
		Default Value:	3h			IPE_3D			
Format:			OpCode						
	26:24	D Command Opcode							
Default Value: 0h				0h 3DST/	ATE_PIPEL	INED			
Foi		Format:	ormat:						
	23:16	3D Command Sub Op	code						
		Default Value:	2Ch 3DSTAT	E_SAMPLE	SAMPLER_STATE_POINTERS_HS				
		Format:	OpCode						
	15:8	Reserved							
		Format:		MBZ					
	7:0	DWord Length					1		
		Default Value:		0h DWORD_COUNT_n					
		Format:		=n					
1	31:5	Pointer to HS Sample	r State				1		
		Format: Dynam	icStateOffset[31:5]SAM	PLER_STA	TE*16			
		Specifies the 32-byte a offset is relative to the	Specifies the 32-byte aligned address offset of the HS function's SAMPLER_STATE table. This offset is relative to the Dynamic State Base Address.						
	4:0	Reserved							
		Format:				MBZ			



		3DSTATE_SA	AMPLEF	R_STA	TE_PO	DINTERS_PS		
Source:		RenderCS						
Length B	lias:	2						
The 3DS	STATE_S	AMPLER_STATE_POINTE	RS_PS comm	and is use	d to define	the location of PS SAMPLER_STATE		
DWord	Bit		atilize sumple	Des	ription			
0	31:29							
		Default Value:			3h G	3h GFXPIPE		
		Format:			OpC	Code		
	28:27	Command SubType						
		Default Value:		3h GFXPIP	E_3D			
		Format:		OpCode				
	26:24	3D Command Opcode						
		Default Value:		0h 3DST/	ATE_PIPELIN	IED		
		Format:	OpCode					
	23:16	3D Command Sub Opd	code					
		Default Value:	2Fh 3DSTAT	e_sample	R_STATE_P	OINTERS_PS		
		Format:						
	15:8	Reserved						
		Format:		MBZ				
	7:0	DWord Length						
		Default Value:		0h DWORD_COUNT_n				
		Format:		=n				
1	31:5	Pointer to PS Sampler	State					
		Format: Dynami	icStateOffset[31:5]SAM	PLER_STATI	E*16		
		Specifies the 32-byte a	ligned addres	s offset o	f the PS fun tress	ction's SAMPLER_STATE table. This		
			Synamic State					
	4:0	Reserved						
		Format:			MBZ			



		3DSTATE_SA	MPLEF	R_STA	TE_F	POINTERS_VS	
Source:		RenderCS					
Length B	ias:	2					
The 3DS table. Or	TATE_S	AMPLER_STATE_POINTE	RS_VS comma utilize sampler	and is use ^r state tab	ed to def oles.	fine the location of VS SAMP	LER_STATE
DWord	Bit			Des	cription		
0	31:29	Command Type					
		Default Value:		3	h GFXPIPE		
		Format:	mat:			DpCode	
	28:27 Command SubType						
	Default Value:				3h GFXI	PIPE_3D	
		Format:			OpCode		
26:24 3D Command Opcode							
		Default Value:			ATE_PIPE	ELINED	
		Format:		OpCode			
	23:16	3D Command Sub Ope	code				
		Default Value:	2Bh 3DSTAT	E_SAMPLER_STATE_POINTERS_VS			
		Format:	OpCode				
	15:8	Reserved					
		Format:				MBZ	
	7:0	DWord Length					
		Default Value:		0h DWORD_COUNT_n			
		Format:		=n			
1	31:5	Pointer to VS Sampler	State				1
		Format: Dynam	icStateOffset[31:5]SAM	PLER_ST	ATE*16	
		Specifies the 32-byte a offset is relative to the I	ligned addres Dynamic State	s offset o Base Ade	f the VS dress.	function's SAMPLER_STATE t	able. This
	4:0	Reserved					
		Format:				MBZ	



			3DS1	ΓΑΤΙ	E_SBE				
Source:		RenderCS							
Length E	Bias:	2							
DWord	Bit		Description						
0	31:29	Command Type							
		Default Value:				3h GFXPIPE			
		Format:				OpCode			
	28:27	Command SubType							
		Default Value:			3h GF	EXPIPE_3D			
		Format:			ode				
	26:24	3D Command Opcode							
		Default Value:		0h 3D	STATE_PI	PELINED			
		Format: OpCo			de				
	23:16	3D Command Sub Opcode)						
		Default Value:			1Fh 3DST	TATE_SBE			
		Format:			OpCode				
	15:8	Reserved							
		Format: MBZ							
	7:0	DWord Length							
		Default Value: 0C			udes DW	/ord (0,1)			
		Format:	=n						
		Total Length - 2							
1	31:29	Reserved							
		Format:				MBZ			
	28	Attribute Swizzle Control Mode							
		Format:	U1 enu	merated	l type				
		When Attribute Swizzle E	nable is	ENABL	ED, this	bit controls whether attributes 0	-15 or		
		16-31 are subject to the f	ollowing	g swizzl	le contro	ols:			
		Attribute n Compone	ent Overr	ide X/Y/	/Z/W				
		Attribute n Constant	Source						
		Attribute n Swizzle Se	elect						
		Attribute n Source At	tribute	blac					
		• Altribute in wrap Sho	ntest Ena	ibles					
		Note that the Number of	SF Outp	out Attr	ibutes fi	eld specifies how many attribute	es are		
		Note: This field does not	impact	any fur	octions	which provide separate states for	all 32		
		attributes (e.g., Point spri	te, Cons	tant inf	erpolati	on).	an sz		
		Value Name			Dese	cription			



			3DSTATE_	SB	E				
	0h	SWIZ_0_15	Attributes 0-15 are subject not.	t to	swizzling, and attributes 16-31 are				
	1h	SWIZ_16_31	IZ_16_31 Attributes 16-31 are subject to swizzling, and attributes 0-15 are not. Only valid when 16 or more attributes are output.						
27:22	Number of SF Output Attributes								
	Forma	Format: U6 count of attributes							
	Specifi	Specifies the number of vertex attributes passed from the SF stage to the WM stage (does not							
	include	e Position).							
	Value				Name				
	[0,32]	[0,32]							
21	Attrib	ute Swizzle Er	nable						
	Forma	at:			Enable				
	attribu	inables the SF to perform swizzling on (up to the first 16) vertex attributes. If DISABLED, all vertex attributes are passed through.							
20	Point	Point Sprite Texture Coordinate Origin							
	Forma	at:	U1 enumerated typ	be					
	This sta	ate controls ho	ow Point Sprite Texture Coc	ordi	nates are generated (when enabled on a pe				
	attribu	te basis by Po	int Sprite Texture Coordina		nable).				
	Oh		Top Left = $(0.0.0.1)$ Bottom Left = $(0.1.0.1)$ Bottom Bight = $(1.1.0.1)$						
	1h		10WERLEFT Top Left = (0,1,0,1)Bottom Left = (0,0,0,1)Bottom Right = (1,0,0,1)						
10.10	Becom	red			(1,0,0,1)Dottom (1,0,0,1)				
19.10	Forma	nt:			MB7				
15:11	Vertex Forma	t URB Entry R at: U5 Specifie register inc	ead Length is the amount of URB data i rements.	read	for each Vertex URB entry, in 256-bit				
			Value		Name				
	[1,16]								
	Programming Notes								
	It is UNDEFINED to set this field to 0 indicating no Vertex URB data to be read. This field should be set to the minimum length required to read the maximum source attribute. The maximum source attribute is indicated by the maximum value of the enabled Attribute # Source Attribute if Attribute Swizzle Enable is set, Number of Output Attributes-1 if enable is not set. read_length = ceiling((max_source_attr+1)/2)								
10	Reserv	/ed							
9:4	Vertex Specifi	URB Entry R es the offset (i	ead Offset n 256-bit units) at which Ve	erte	x URB data is to be read from the URB.				
3:0	Reserv	/ed							



		3[DST	ATE_SBE							
	Forma	at:		MBZ							
31	1 Attrib	ute [2n+1] Component	Overr	ide W							
	Forma	ət:		Enable							
	If set, t vector	he W component of outp specified by ConstantSou	out Att urce[1]	tribute 1 is overridden by the W component of th].	e constar						
30	Attrib	Attribute [2n+1] Component Override Z									
	Forma	at:		Enable							
	If set, t vector	If set, the Z component of output Attribute 1 is overridden by the Z component of the constant vector specified by ConstantSource[1].									
29	9 Attrib	ute [2n+1] Component	Overr	ide Y							
	Forma	at:		Enable							
	If set, t vector	f set, the Y component of output Attribute 1 is overridden by the Y component of the constant vector specified by ConstantSource[1].									
28	3 Attrib	ute [2n+1] Component	Overr	ide X							
	Forma	at:		Enable							
	If set, t vector	he X component of outp specified by ConstantSou	ut Attr urce[1]	ribute 1 is overridden by the X component of the].	constant						
27	7 Reserv	Reserved									
	Forma	at:		MBZ							
26:2	25 Attrib	ute [2n+1] Constant So	urce								
	Forma	at: U2	2 enum	nerated type							
	This st Attribu	ate selects a constant veo ite 1	ctor wł	nich can be used to override individual compone	nts of						
	Valu	e Name		Description							
	0h	CONST_0000		Constant.xyzw = 0.0,0.0,0.0,0.0							
	1h	CONST_0001_FLOAT		Constant.xyzw = 0.0,0.0,0.0,1.0							
	2h	CONST_1111_FLOAT		Constant.xyzw = 1.0,1.0,1.0,1.0							
	3h	PRIM_ID		Constant.xyzw = PrimID (replicated)							
24	4 Reserv	/ed									
	Forma	at:		MBZ							
23:2	22 Attrib	ute [2n+1] Swizzle Sele	ct								
	Forma	at: U2	enum	nerated type							
	This st	ate, along with Attribute	1 Sour	rce Attribute, specifies the source for output Attri	bute 1.						
	Value	Name		Description							
	0h	INPUTATTR	This AttrI	attribute is sourced from nputReg[SourceAttribute]							
	1h	INPUTATTR FACING	If the	e obiect is front-facing, this attribute is sourced							



		30	STATE_SB	E				
			from AttrInputReg[back-facing, this att AttrInputReg[Sourc	Source tribute eAttrik	eAttribut is sourc oute+1].	te]. If the object is ced from		
	2h	INPUTATTR_W	This attribute is sou AttrInputReg[Sourc copied to the X con	irced f eAttrik npone	rom oute]. Th nt.	ne W component is		
	3h	INPUTATTR_FACING_W	If the object is front from AttrInputReg[back-facing, this att AttrInputReg[Sourc copied to the X con	t-facin Source tribute eAttrik npone	g, this a Attribut is sourc oute+1]. nt.	ttribute is sourced te]. If the object is ced from . The W component is		
21	Reserv	Reserved						
	Forma	at:			MBZ			
20:16	Attrib	ute [2n+1] Source Attrib	oute					
	Forma	at:				U5		
	This fie 128 bit	This field selects the source attribute for Attribute 1. Source attribute 0 corresponds to the first 128 bits of data indicated by Vertex URB Entry Read Offset						
15	Attribute [2n] Component Override W							
	Forma	at:		Enable	è			
	If set, the W component of output Attribute 0 is overridden by the W component of the const vector specified by ConstantSource[1].					constant		
14	Attrib	ute [2n] Component Ove	erride Z					
	Forma	at:		Enable	è			
	If set, t vector	the Z component of outpu specified by ConstantSou	it Attribute 0 is overi rce[1].	ridden	by the 2	Z component of the co	onstant	
13	Attrib	ute [2n] Component Ove	erride Y					
	Forma	at:		Enable				
	If set, t vector	the Y component of outpu specified by ConstantSou	ut Attribute 0 is overn rce[1].	ridden	by the `	Y component of the co	onstant	
12	Attrib	ute [2n] Component Ove	erride X					
	Forma	at:		Enable	9			
	If set, t vector	the X component of outpu specified by ConstantSou	ut Attribute 0 is overi rce[1].	ridden	by the	X component of the co	onstant	
11	Reserv	ved						
	Forma	at:			MBZ			
10:9	Attrib	ute [2n] Constant Source	e					
	Forma	at: U2	enumerated type					
	This state selects a constant vector which can be used to override individual components of					s of		



			3D	ST	ATE_SBE			
		Attribute	e 0					
		Value	Name		De	scriptio	n	
		0h	CONST_0000		Constant.xyzw = 0.0,0.0,0.0,0.0			
		1h	CONST_0001_FLOAT		Constant.xyzw = 0.0,0.0,0.0,1.0			
		2h	CONST_1111_FLOAT		Constant.xyzw = 1.0	,1.0,1.0,1	1.0	
		3h	PRIM_ID		Constant.xyzw = Pri	mID (rep	olicated)	
	8	Reserve	d					
		Format				MBZ		
	7:6	Attribut	te [2n] Swizzle Select					
		Format:	: U2	enum	nerated type			
		This stat	e, along with Attribute 0) Sour	ce Attribute, specifies	s the sou	urce for output Attribu	ute 0.
		Value	Name		Dese	cription		
		0h	INPUTATTR	This AttrI	attribute is sourced finnputReg[SourceAttrik	rom oute]		
	1h INPUTATTR_FACING If the from back-AttrIr 2h INPUTATTR_W This a AttrIr copie		If the object is front-facing, this attribute is sourced from AttrInputReg[SourceAttribute]. If the object is back-facing, this attribute is sourced from AttrInputReg[SourceAttribute+1].					
			This AttrIi copie	This attribute is sourced from AttrInputReg[SourceAttribute]. The W component is copied to the X component.				
		3h	INPUTATTR_FACING_W	If the from back AttrI copie	e object is front-facing AttrInputReg[Source -facing, this attribute nputReg[SourceAttrik ed to the X compone	g, this at Attribut is sourc oute+1]. nt.	ttribute is sourced e]. If the object is red from The W component is	
	5	Reserve	d					-
		Format	:			MBZ		
	4:0	Attribut	te [2n] Source Attribute	e				
		Format:	:				U5	
		This field selects the source attribute for Attribute 0. Source attribute 0 corresponds to the 128 bits of data indicated by Vertex URB Entry Read Offset						ne first
10	31:0	Point Sp	prite Texture Coordinat	te Ena	able			
		Format	:	32	2-bit bitmask			
					Description			
		When	processing point prim	itives	the attributes from	n the in	comina point verte	x are
When processing point primitives, the attributes from the incoming p typically copied to the point object corner vertices. However, if a bit i the corresponding Attribute is selected as a Point Sprite Texture Coo						r, if a bit is set in this ture Coordinate, in	s field, which	



		3DSTATE_SBE
		the Point Sprite Texture Coordinate Origin state bit. Bit 0 corresponds to output Attribute 0.
		This field must be programmed to 0 when non-point primitives are rendered.
11	31:0	Constant Interpolation Enable[31:0] This field is a bitmask containing a Constant Interpolation Enable bit for each corresponding attribute. If a bit is set, that attribute will undergo constant interpolation, and the corresponding WrapShortest Enable bits (if defined) will be ignored. If a bit is clear, components which are not enabled for WrapShortest interpolation (if defined) will be linearly interpolated.
12	31:28	Attribute 7 WrapShortest Enables
		Format: Enable[4]
		This state selects which components (if any) of Attribute 7 are to be interpolated in a "wrap shortest" fashion. Operation is UNDEFINED if any of these bits are set and the Constant Interpolation Enable bit associated with this attribute is set. Note that wrap-shortest interpolation is only supported for Attributes 0-15. Bit 0: WrapShortest X ComponentBit 1: WrapShortest Y ComponentBit 2: WrapShortest Z ComponentBit 3: WrapShortest W Component
	27:24	Attribute 6 WrapShortest Enables (See above).
	23:20	Attribute 5 WrapShortest Enables (See above).
	19:16	Attribute 4 WrapShortest Enables (See above).
	15:12	Attribute 3 WrapShortest Enables (See above).
	11:8	Attribute 2 WrapShortest Enables (See above).
	7:4	Attribute 1 WrapShortest Enables (See above).
	3:0	Attribute 0 WrapShortest Enables (See above).
13	31:28	Attribute 15 WrapShortest Enables
		Format: Enable[4]
		This state selects which components (if any) of Attribute 15 are to be interpolated in a "wrap shortest" fashion. Operation is UNDEFINED if any of these bits are set and the Constant Interpolation Enable bit associated with this attribute is set.Bit 0: WrapShortest X ComponentBit 1: WrapShortest Y ComponentBit 2: WrapShortest Z ComponentBit 3: WrapShortest W Component
	27:24	Attribute 14 WrapShortest Enables
		(See above).
	23:20	Attribute 13 WrapShortest Enables (See above).
	19:16	Attribute 12 WrapShortest Enables



	3DSTATE_SBE						
	(See above).						
15:12	Attribute 11 WrapShortest Enables (See above).						
11:8	Attribute 10 WrapShortest Enables (See above).						
7:4	Attribute 9 WrapShortest Enables (See above).						
3:0	Attribute 8 WrapShortest Enables (See above).						

Г



		3DST	ATE_S	SCISSO	DR_ST		E_POIN1	ERS	
Source:		RenderCS							
Length Bias: 2									
The 3DSTATE_SCISSOR_STATE_POINTERS command is used to define the locat state.					e the location o	f the indired	ct SCISSOR_RECT		
DWord	Bit		Description						
0	31:29	Command Type	Command Type						
		Default Value:				3h GFXPIPE			
		Format:					OpCode		
	28:27	Command Sub	Туре						
		Default Value:		3h GF	XPIPE_3D				
		Format:				OpCode			
	26:24	3D Command Opcode							
		Default Value:	Default Value:			ATE_PI	PELINED		
		Format:			OpCode				
	23:16	3D Command Sub Opcode							
		Default Value: 0Fh 3DSTA			TE_SCISSOR_STATE_POINTERS				
		Format:		OpCode					
	15:8	Reserved							
		Format:			MBZ				
	7:0	DWord Length							
		Default Value:			0h DWC	0h DWORD_COUNT_n			
		Format:			=n				
1	31:5	Scissor Rect Po	inter						
		Format:	Dynamic	StateOffset	[31:5]SCIS	SOR_R	ECT*16		
		Specifies the 32-byte aligned address offset of the SCISSOR_RECT state. This offset is						This offset is	
		relative to the	Dynamio	c State Bas	se Addre	SS			
	4:0	Reserved							
		Format:					MBZ		



	3DSTATE_SF								
Source:		Ren	derCS						
Length Bias: 2									
DWord	Bit				Des	cripti	on		
0	31:29	Command	d Type						
		Default Value: 31					3h GFXPIPE		
		Format:					OpCode		
	28:27	Command	l SubType			-			
		Default V	alue:			3h G	FXPIPE_3D		
		Format:				OpC	ode		
	26:24	3D Comm	and Opcode						
		Default V	alue:				0h 3DSTATE		
		Format:					OpCode		
	23:16	3D Comm	and Sub Opcode						
		Default V	alue:		1	.3h 3D	DSTATE_SF		
		Format:			C	DpCoc	de		
	15:8	Reserved							
		Format:	Format: MBZ						
	7:0	DWord Length							
		Default V	alue:		5h Exclud	es DV	Vord (0,1)		
		Format:			=n Total	Lengtl	h - 2		
1	31:15	Reserved							
		Format:					MBZ		
	14:12	Depth But	ffer Surface Forn	nat					
		Format:		U3 Enur	merated Ty	уре			
		Specifies t	Specifies the format of the depth buffer. This must exactly match the Surface Format						
		Global Der	oth Bias.		UFFER. III	e sr n	equires this information in orde	r to compute	
		Value	Na	ame			Description	1	
		0h	D32_FLOAT_S8X2	24_UINT		D32	_FLOAT_S8X24_UINT		
		1h	D32_FLOAT			D32_	_FLOAT		
		2h	D24_UNORM_S8	_UINT		D24	_UNORM_S8_UINT		
		3h	D24_UNORM_X8	_UINT		D24	_UNORM_X8_UINT		
		4h Reserved				Reserved			
	5h D16_UNORM				D16_UNORM				
		6h-7h Reserved Reserved				erved			
	11	Legacy Gl	obal Depth Bias	Enable					
		Format:					Enable		
		Enables th	e SF to use the GI	obal Dep	oth Offset	Const	ant state unmodified. If this bit	is not set, the	



			3DSTATE_SF							
	SF will s found.	scale the Glob of this docum	al Depth Offset Constant as described in section Error! Reference so ent.	ource not						
	Programming Notes									
	This bit should be set whenever non zero depth bias (Slope, Bias) values are used. Setting the bit may have some degradation of performance for some workloads.									
10	Statisti	Statistics Enable								
	Forma	t:	Enable							
	If ENAB DISABL	BLED, this FF u ED, CL_PRIMI	nit will increment CL_PRIMITIVES_COUNT on behalf of the CLIP stag IVES_COUNT will be left unchanged.	e. If						
			Programming Notes							
	This bi CLIP_S clear.	This bit should be set whenever clipping is enabled and the Statistics Enable bit is set in CLIP_STATE. It should be cleared if clipping is disabled or Statistics Enable in CLIP_STATE is clear.								
9	Global	Depth Offset	Enable Solid							
	Forma	t:	Enable							
	Enables	Enables computation and application of Global Depth Offset for SOLID objects.								
	Programming Notes									
	This bi Setting	This bit should be set whenever non zero depth bias (Slope, Bias) values are used. Setting this bit may have some degradation of performance for some workloads.								
8	Global Depth Offset Enable Wireframe									
	Forma	t:	Enable							
	Enables WIREFF	Enables computation and application of Global Depth Offset when triangles are rendered in WIREFRAME mode.								
	Programming Notes									
	This bit should be set whenever non zero depth bias (Slope, Bias) values are used. Setting this bit may have some degradation of performance for some workloads.									
7	Global	Depth Offset	Enable Point							
	Forma	t:	Enable							
	Enables POINT	Enables computation and application of Global Depth Offset when triangles are rendered in POINT mode.								
			Programming Notes							
	This bi Setting	This bit should be set whenever non zero depth bias (Slope, Bias) values are used. Setting this bit may have some degradation of performance for some workloads.								
6:5	FrontFa	ace Fill Mode		<u> </u>						
	Forma	t:	U2 enumerated type							
	This sta	te controls ho	w front-facing triangle and rectangle objects are rendered.							
	Value	Name	Description							
	0h	0h SOLID Any triangle or rectangle object found to be front-facing is rendered as a solid object. This setting is required when rendering rectangle (RECTUST) objects								
	1h	WIREFRAME	 Any triangle object found to be front-facing is rendered as a series of lines along the triangle boundaries (as determined by 							



				3DSTATE_SF						
				the topology type and controlled by the vertex EdgeFlags).						
		2h	POINT	Any triangle object found to be front-facing is rendered as a set of point primitives at the triangle vertices (as determined by the topology type and controlled by the vertex EdgeFlags). NOTE: If the triangle is clipped, points will not be rendered at clip-inserted vertices. Point will only be rendered at original vertices (if visible).						
		3h	Reserved							
	4:3	BackFa	 Face Fill Mode							
		Format: U2 enumerated type								
		This sta	te controls ho	w back-facing triangle and rectangle objects are rendered.						
		Value	Name	Description						
		0h	SOLID	Any triangle or rectangle object found to be back-facing is rendered as a solid object. This setting is required when rendering rectangle (RECTLIST) objects.						
		1h	WIREFRAME	Any triangle object found to be back-facing is rendered as a series of lines along the triangle boundaries (as determined by the topology type and controlled by the vertex EdgeFlags).						
2h POINT Any triangle object found of point primitives at the topology type and contro the triangle is clipped, point vertices. Point will only be				Any triangle object found to be back-facing is rendered as a set of point primitives at the triangle vertices (as determined by the topology type and controlled by the vertex EdgeFlags). NOTE: If the triangle is clipped, points will not be rendered at clip-inserted vertices. Point will only be rendered at original vertices (if visible).	to be back-facing is rendered as a set triangle vertices (as determined by the lled by the vertex EdgeFlags). NOTE: If ints will not be rendered at clip-inserted e rendered at original vertices (if visible).					
		3h	Reserved							
	2	Reserve	ed							
		Format	t:	MBZ						
	1	View T	ransform Ena	ble						
		Format	t:	Enable						
		This bit	controls the V	Viewport Transform function.						
	0	Front V Determ position winding	Vinding ines whether a ns, when trave g order. Does r	a triangle object is considered "front facing" if the screen space vertex ersed in the order, result in a clockwise (CW) or counter-clockwise (CCW) not apply to points or lines.						
2	31	Anti-Al	liasing Enable	2						
		Format	t:	Enable						
		This fiel	ld enables "alp	oha-based" line anti-aliasing.						
				Programming Notes						
		This fie format	eld must be dis	sabled if any of the render targets have integer (UINT or SINT) surface						
	30:29	Cull Mo	ode							
		Format	t:	3D_CullMode						
		Controls removal (culling) of triangle objects based on orientation. The cull mode only applies to								



				3DST	ATE_SI	F			
	triangle objects and does not apply to lines, points or rectang								
	Value	Name	e			Descr	ription		
	0h	CULLMODE	BOTH	All triangles are discarded (i.e., no triangle objects are drawn)					
	1h	CULLMODE	NONE	No triang	les are discar	ded du	ue to orientation		
	2h	CULLMODE	FRONT	Triangles with a front-facing orientation are discarded					
	3h	CULLMODE_	ВАСК	Triangles	with a back-f	facing	orientation are d	liscarded	
				P	Programmin	g Note	es		
	Orient	ation determi	ination i	s based on	the setting o	of the l	Front Winding st	ate.	
28	Reserv	ed							
27:18	Line W	idth							
	Forma	t:					U3.7		
	Range	: [0.0, 7.99218	375]						
	Contr	ols width of	line pri	mitives. Se	etting a Line	e Widt	h of 0.0 specifi	es the rasterizati	
	of the "thinnest" (one-pixel-wide), non-antialiased lines. Note that this effectively								
	overrides the effect of AAEnable (though the AAEnable state variable is not modified).								
		Programming Notes							
	Software must not program a value of 0.0 when running in MSRASTMODE_ON_xxx								
	modes - zero-width lines are not available when multisampling rasterization is								
	enable	eu.							
17:16	Line End Cap Antialiasing Region Width								
	Format: U2								
	compu	This field specifies the distances over which the coverage of anti-aliased line end caps are computed							
		/alue	Na	ame	C	Descrip	otion		
	0h				0.5 pixels				
	1h				1.0 pixels				
	2h				2.0 pixels				
	3h				4.0 pixels				
15	Reserv	ed			•			<u>.</u>	
	Forma	t:					MBZ		
14	Reserv	ed					•		
	Format: MBZ								
13	Reserv	ed							
12	Reserv	ed							
11	Scissor	Rectangle E	nable						
	Forma	t:				Enable	e		



				3DST	ATE_SF						
		Enables operation of	of Scisso	or Rectangle.							
	10	Reserved									
		Format:				MBZ					
	9:8	Multisample Rasterization Mode									
		Format: U2 enumerated type									
		This state is duplica 3DSTATE_WM for c	ited in 3 lefinitio	DSTATE_WN n details.	A and both must be	set to the sam	e value. See the field in				
	7:0	Reserved									
		Format:		MBZ							
3	31	Last Pixel Enable									
		Format:			Enable	•					
		If ENABLED, the las	t pixel o	of a diamond	l line will be lit. This	state will only	affect the rasterization				
		of Diamond lines (v	vill not a	affect wide li	nes or anti-aliased l	ines).					
		Programming Notes									
		Last pixel is applied to all lines of a LINELIST, and only the last line of a LINESTRIP.									
	30:29	Triangle Strip/List									
		Format:	Format: 0-based vertex index								
		vertex". Used for flat shading of primitives. Does current implementation send provoking first?									
		Value			Name						
		0h		Vertex 0							
		1h		Vertex 1							
		2h		Vertex 2							
		3h		Reserved							
	28:27	Line Strip/List Pro	vokina	Vertex Sele	ect						
		Format:	- -	0-based ve	ertex index						
		Selects which verte	x of a lir	ne (in a line s	strip or list primitive) is considered	the "provoking vertex".				
		Value	1	Name	Descrip	tion					
		0h			Vertex 0						
		1h			Vertex 1						
		2h			Reserved		_				
		3h			Reserved						
	26:25	Triangle Fan Prove	okina V	ertex Select	<u> </u>						
		Format:	<u>-</u>	0-based ve	ertex index						
		Selects which verte	x of a tr	iangle (in a t	riangle fan primitive	e) is considered	the "provoking vertex".				
		Value			Name						
		0h		Vertex 0							



				3	DSTATE_SF			
		1h		Vert	tex 1			
		2h		Vert	tex 2			
		3h		Rese	erved			
	24:15	Reserved	Reserved					
		Format:			MBZ			
	14	AA Line [Distance Mod	le				
		Format:			U1			
		This bit cc	ontrols the dis	tance co	omputation for antialiased lines.			
		Value	Name		Description			
		0h Re	eserved		Reserved			
		1h A	ALINEDISTAN	CE_TRUE	E True distance computation. This is the normal which should yield WHQL compliance.	setting		
	13	Reserved						
		Format:			MBZ			
	12	Vertex Su	ub Pixel Preci	ision Sel	lect			
		Format:			U1			
		Selects the	e number of f	ractional	I bits maintained in the vertex data			
		Value	Name		Description			
		0h	Disable	8 sub	pixel precision bits maintained			
		1h	Enable	4 sub	pixel precision bits maintained			
	11	Use Point	t Width State	,				
		Format:	1					
		Controls v primitives	vhether the p	oint widt	th passed on the vertex or from state is used for	rendering point		
		Value	Name	e	Description			
		0h		Use	e Point Width on Vertex	_		
		1h		Use	e Point Width from State			
	10:0	Point Wig	dth					
		Format:			U8.3			
		Range: [0).125, 255.875] pixels				
		This field specifies the size (width) of point primitives in pixels. This field is overridden (though not overwritten) whenever point width information is passed in the FVF						
4	31:0	Global De	epth Offset C	onstant	t			
		Format:			IEEE_FP			
		Specifies t	the constant t	erm in th	he Global Depth Offset function.			
-	21.0							
5	31:0	Global De	epth Offset S	cale				
		Specifies t	the scale term	used in	the Global Depth Offset function			
		Specifics (the scale term	ruscu m	The diobal Depth Onset function.			



	3DSTATE_SF						
6	31:0	Global Depth Offset Clamp					
		Format:	IEEE_FP				
		Specifies the clamp term used in the Global Depth Offset function.					



3DSTATE SO BUFFER										
Source:		RenderCS								
Length E	Bias:	2								
DWord	Bit	Description								
0	31:29	Command Type								
		Default Value:		3h GFXPIPE						
		Format:		OpCode						
	28:27	Command SubType								
		Default Value:		3h GFXPIPE_3D						
		Format:				OpCode				
	26:24	3D Command Opcode								
		Default Value:	1h			3DSTATE_NONPIPELINED				
		Format:	Ор	Code	e					
	23:16	3D Command Sub Opcode								
		Default Value:	1	18h 3DSTATE_SO_BUFFER						
		Format:	(OpCode						
	15:8	Reserved								
		Format: MBZ								
	7:0	DWord Length								
		Default Value:		2h Excludes DWord (0,1)						
		Format:			=n					
		Total Length - 2								
1	31	Reserved								
		Format:	MBZ							
	30:29	SO Buffer Index								
		Format: U2								
		Specifies which of the four SO Buffers is being defined.								
	28:25	SO Buffer Object Control State								
		Format: MEMORY_OBJECT_CONTROL_STATE								
		Specifies the memory object control state for the SO buffer.								
	24:22	Reserved								
		Format:		MBZ						
	21:12	Reserved							1	
		Format:			MBZ					
	11:0	Surface Pitch								
		Format: U12 Pitch in Bytes								
		This field specifies the pitch of the SO buffer in #Bytes.								


			3DSTATE_SO_BUFFE	R					
		Value	Na	ame					
		[0,2048]	Must be 0 or a multiple of 4 Bytes.						
		Programming Notes							
		A Surface Pitch of 0 indicates an un-bound buffer. No writes are performed. Surface Base Address is ignored.							
2	31:2	Surface Base Address							
		Format:	GraphicsAddress[31:2]						
		This field specifies the	ne starting DWord address LSBs of the	buffer in Graphics Memory.					
	1:0	Reserved							
		Format:		MBZ					
3	31:2	Surface End Addres	55						
		Format:	GraphicsAddress[31:2]						
This field specifies the ending DWord address of the buffer in Graphics Memory.									
	1:0	Reserved							
		Format:		MBZ					



			3DSTAT	E_SO_D	ECL_I	IST				
Source:		Ren	derCS							
Length E	Bias:	2	2							
DWord	Bit	Description								
0	31:29	Command	l Туре							
		Default V	alue:		3h	GFXPIPE				
		Format:			OpCode					
	28:27	Command	l SubType							
		Default V	alue:		3h GFXPI	PE_3D				
		Format:			OpCode					
	26:24	3D Comm	and Opcode							
		Default Va	alue:	1h 3DSTATE_	NONPIPEL	INED				
		Format:		OpCode						
	23:16	3D Comm	and Sub Opcode	T						
		Default V	alue:	17h 3DSTATE	SO_DECL	_LIST				
		Format:		OpCode						
	15:9	Reserved								
_		Format:				MBZ				
	8:0	DWord Length								
		Format:	=r	n Total Length	- 2					
			Fo	ormat: O1						
			I							
		Value	N	ame		Description				
		3h	Excludes DWord (0,1)	[Default]	Default value = $2(N-1)+3$ h					
1	31:16	Reserved								
		Format:				MBZ				
	15:12	Stream to	Buffer Selects [3]							
		Format:	U4	bitmask						
			In	dex of SO Stre	am					
		Identifies t	o which SO Buffers str	ream 3 outputs	. See Strea	am To Buffer Selects [0] field description.				
		racialités le miler de danées succiri s outpuis, dee ducarri to danée delecis [6] rela description.								
	11:8	Stream to	Buffer Selects [2]							
		Format:		U4 bit	tmask					
		Identifies t	to which SO Buffers str	eam 2 outputs	. See Strea	am To Buffer Selects [0] field description.				
	7:4	Stream to	Buffer Selects [1]							
		Format:		U4 bit	tmask					
		Identifies t	o which SO Buffers str	eam 1 outputs	. See Strea	am To Buffer Selects [0] field description.				



		3DSTA	ſE_SO_DE	CL_LIST				
	3:0	Stream to Buffer Selects [0]	t					
		Format:	U4 bitm	lask				
		Identifies to which SO Buffers stream 0 outputs (irrespective of whether those buffers are enabled via 3DSTATE_STREAMOUT). Software is required to scan the SO_DECL list in order to provide this summary information. Note: For "inactive" streams, software must program this field to all zero (no buffers written to) and the corresponding Num Entries field to zero (no valid SO_DECL s).						
		Value		Name				
		1xxxb	SO Buffer 3					
		x1xxb	SO Buffer 2					
		xx1xb	SO Buffer 1					
		xxx1b	SO Buffer 0					
2	31:24	Num Entries [3]						
		Format:	U8 #ent	ries				
		Specifies the number of valid SO_DECL entries for Stream 3. (See notes in Num Entries [0] field description).						
		Value		Name				
		[0,128]		entries				
	23:16	Num Entries [2]						
		Format:	U8 #ent	ries				
		Specifies the number of valid SC description).)_DECL entries fo	r Stream 2. (See notes in Num Entries [0] field				
		Value		Name				
		[0,128]		entries				
	15:8	Num Entries [1]						
		Format: U8 #entries						
		Specifies the number of valid SC description).)_DECL entries to	r Stream 1. (See notes in Num Entries [0] field				
		Value		Name				
		[0,128]		entries				
	7:0	Num Entries [0]						
		Format:	U8 #ent	ries				
		Specifies the number of valid SO_DECL entries for Stream 0.Note that the SO_DECLs are programmed in groups of four (one SO_DECL for each of the four streams). Therefore the number of 2-DWord groups of SO_DECLs supplied in this command is derived from the stream with the most valid SO_DECLs. The NumEntries value specific to each stream will indicate how many SO_DECLS are valid for that particular stream. Any trailing invalid SO_DECLs supplied for streams with fewer valid SO_DECLs will be ignored. It is legal to specify Num Entries = 0 for all four streams simultaneously. In this case there will be no SO_DECLs included in the command (only DW 0-2). Note that all Stream to Buffer Selects bits must be zero in this case (as no strea						
		Value		Name				

г



	3DSTATE_SO_DECL_LIST									
		[0,128]	entries							
3n	63:48	SO_DECL[3,n]								
		Format:	SO_DECL							
		This field contains Stream 3 SO_DECL [n]								
	47:32	SO_DECL[2,n]								
		Format:	SO_DECL							
		This field contains Stream 2 SO_DECL [n]								
	31:16	SO_DECL[1,n]								
		Format:	SO_DECL							
		This field contains Stream 1 SO_DECL [n]								
	15:0	SO_DECL[0,n]								
		Format:	SO_DECL							
		This field contains Stream 0 SO_DECL [n]								

Doc Ref # IHD-OS-VLV-Vol2 pt2-04.14



			3DSTATI	E_S ⁻	TENC	IL_E	BUFFER		
Source:		RenderC	S						
Length B	ias:	2							
This com However	imand s ; the st	sets the surface ate change pip	state of the sep elining isn't com	arate s pletely	stencil buf / transpare	fer, de ent (se	elivered as a pipelined state comma ee restriction below).	and.	
			Progr	ammi	ing Notes				
Restricti 3DSTAT 3DSTAT Stall bit followed otherwis MI_FLUS 3DSTAT	on: Pric E_DEPT E_HIER_ set, fol d by an se guar SH). E_STEN	or to changing 'H_BUFFER, 3DS _DEPTH_BUFFE lowed by a pipe other pipelined antee that the ICIL_BUFFER mu	Depth/Stencil Bu STATE_CLEAR_PA R) SW must first elined depth cacl depth stall (PIPE pipeline from WI	Iffer st RAMS issue a he flus E_CON M onw	ate (i.e., an 5, 3DSTATE a pipelined sh (PIPE_Co ITROL with vards is alr med in the	ny con E_STEN d dept ONTR(n Dept eady f	nbination of NCIL_BUFFER, h stall (PIPE_CONTROL with Depth OL with Depth Flush Bit set, th Stall Bit set), unless SW can flushed (e.g., via a preceding g with the other Depth/Stencil	_	
state co 3DSTAT	mmano E HIER	ls(i.e. 3DSTATE <u>)</u> DEPTH_BUFFE	_DEPTH_BUFFER, R)	3DST	ATE_CLEA	R_PAR	RAMS, or		
The ster	ncil buf	fer is always Til	e-Y					-	
DWord	Bit	Description							
0	31:29	Command Type							
		Default Value:					3h GFXPIPE		
		Format:					OpCode		
	28:27	Command SubType							
		Default Value: 3h G					EXPIPE_3D		
		Format: OpCode							
	26:24	3D Command Opcode							
		Default Value:			0h 3DSTA)h 3DSTATE_PIPELINED			
		Format:			OpCode				
	23:16	3D Command	Sub Opcode						
		Default Value:		06h 3	DSTATE_S	E_STENCIL_BUFFER			
		Format:		ОрСо	ode				
	15:8	Reserved							
		Format:					MBZ		
	7:0	Dword Length	1						
		Format:	=	n Tot	al Length	- 2			
		Value			Na	mo			
		1h	Excludes Dword	(0.1)	[Default]	iiie			
1	31	Reserved		(-,-)					
1	51	Format:					MBZ		
	30:29	Reserved					1		



			BDS	A	TE_STENCIL_BU	FFER			
		Format:				MBZ	٦		
	28:25	Stencil Buffer Object Control State							
		Format:	MEN	/OR	COBJECT_CONTROL_STATE				
		Description							
		Specifies the memory object control state for the stencil buffer. Stencil Buffer Object Control State [3:0]							
		This field is not	t contex	t save	e and restored by hardware. It	f this field is programmed to			
		events:		ero, n	t must be programmed after	the following commands of			
		 MI_SET_ 	CONTEX	(T					
		• MI_WAI	T_FOR_E	VEN	T (Specifically waits on vblank	or display flip)			
		Render	engine o	joes	IDLE due to head point equal	to tail pointer			
	24:22	Reserved							
		Format:				MBZ			
	21:17	7 Reserved							
		Format:				MBZ			
	16:0	Surface Pitch			1				
		Format:			U17-1 Pitch in Bytes				
		This field specif	ies the p	itch	of the stencil buffer in (#Byte	s - 1).	-		
		Value	Name		Des	scription			
		[127, 3FFFFh]		corr	esponding to [128B, 128KB]a	lso restricted to a multiple of 128B			
					Programming Not	es	1		
		Since this surfa	ice is tile	d, th	e pitch specified must be a m	nultiple of the tile pitch, in the range	_		
		[128B, 128KB].							
		The pitch must be set to 2x the value computed based on width, as the stencil buffer is stored with two rows interleaved. For details on the separate stencil buffer storage format in memory.							
		see GPU Overv Lavout (sectior	iew (vol 1 8.20.4.8	1a), N 3).	Memory Data Formats, Surface	e Layout, 2D Surfaces, Stencil Buffer			
2	31:0	Surface Base A	ddress						
		Format:	Gra	ohics	Address[31:0]Stencil_Buffer				
		This field specif	ies the s	tartir	ng Dword address of the buff	er in mapped Graphics Memory.			
					Programming Not	es			
		The Stencil Buf	fer can o	only l	be mapped to Main Memory	(uncached).			



		3DSTAT	E_ST	RE	AM	OU	JT			
Source:		RenderCS								
Length B	lias:	2								
This com	mand	contains pipelined state required by	y the SC)L un	it.					
DWord	Bit			Desc	riptior	า				
0	31:29	Command Type								
		Default Value:					3h GFXPIPE			
		Format:	(OpCode						
	28:27	Command SubType								
		Default Value: 3				(PIPE_	_3D			
		Format:			OpCod	le				
	26:24	3D Command Opcode								
		Default Value:	DSTA	ATE_PIP	ELINE	D				
		Format:	ode							
	23:16	3D Command Sub Opcode							1	
		Default Value:	E_STRE	ΑΜΟι	UT					
		Format:	OpCode	е						
	15:8	Reserved								
		Format:				MBZ				
	7:0	DWord Length								
		Default Value:							1h	
		Format:							=n	
		Total Length - 2								
1	31	SO Function Enable						h		
		Format:						U1		
		It set, the SO function is enabled. Vertex data will be streamed out to memory (subject to overflow detection) as controlled by the various SO-related state variables. If clear, the SO function is disabled, and therefore no vertex data will be streamed out to memory. However, the Rendering Disable and Render Stream Select fields will still be used to determine which vertices (if any) are forwarded down the pipeline for (possible) rendering.								
	30	Rendering Disable						h		
		Format:U1If set, the SO stage will not forward any topologies down the pipeline. If clear, the SO stage will forward topologies associated with Render Stream Select down the pipeline. This bit is used even if SO Function Enable is DISABLED.								
	29	Reserved								
	-	Format:				Ν	MBZ			
	28:27	Render Stream Select								



3DSTATE_STREAMOUT

	Format	t:		U2				
			Description					
	Description This field specifies which stream has been selected to be forwarded down the pipeline for possible rendering. Topologies from other streams will not be passed down the pipeline. If Rendering Disable is set, this field is ignored, as no topologies are sent down the pipeline. This high block							
26	Reorde This bit	er Mode controls	how vertices of triangle objects in TRISTRI	IP[_ADJ] and TRISTRIP_REV are				
	reordered for the purposes of stream-out only (does not impact rendering). See table in In							
	Value	Name	Descriptio	n				
	0h	LEADING	Reorder the vertices of alternating trian such that the leading (first) vertices are at v0. A similar reordering is performed TRISTRIP_REV.	ngles of a TRISTRIP[_ADJ] in consecutive order starting on alternating triangles in a				
	1h	TRAILINGReorder the vertices of alternating triangles of a TRISTRIP[_ADJ] such that the trailing (last) vertices are in consecutive order starting at v2. A similar reordering is performed on alternating triangles in a TRISTRIP_REV.						
25	SO Statistics Enable							
	Format	t:	Enable	2				
	This bit	controls	whether StreamOutput statistics register(s	s) can be incremented.				
	Value	Name	Description	1				
	0h	Disable	SO_NUM_PRIMS_WRITTEN[03] and SO_F registers cannot increment.	PRIM_STORAGE_NEEDED[03]				
	1h	Enable	SO_NUM_PRIMS_WRITTEN[03] and SO_F registers can increment.	PRIM_STORAGE_NEEDED[03]				
24:23	Reserve	ed			_			
	Format	t:		MBZ				
22:12	Reserve	ed						
	Format	t:		MBZ				
11	SO Buf	fer Enab	e [3]					
	Format	t:		U1				
	(See SO) Buffer E	nable [0])					
10	SO Buf	fer Enab	e [2]					
	Format	t:		01				
	(See SO	Buffer E	nadie [U])					
9	SO Buf	fer Enab	e [1]					



		3DSTATE_STREAMOU	JT						
		Format:	U1						
		(See SO Buffer Enable [0])							
	8	SO Buffer Enable [0]							
		Format:	UI Ruffer 0 is considered "not bound"						
		and effectively treated as a zero-length buffer for the purpodetection. If an enabled stream's Stream to Buffer Selects in overflow condition. That stream will cause no writes to occu SO_PRIM_STORAGE_NEEDED[<stream>] will increment. This is DISABLED.</stream>	burler of SC output and overflow includes this buffer it is by definition an ur, and only s bit is ignored if SO Function Enable						
	7:0	Reserved							
		Format:	MBZ						
2	31:30	Reserved							
		Format:	MBZ						
	29	Stream 3 Vertex Read Offset							
		Format: U1 count of 256-bit units							
		Specifies amount of data to skip over before reading back S	Stream 3 vertex data.						
		(See Stream 0 Vertex Read Offset)							
	28:24	Stream 3 Vertex Read Length							
		Format: U5-1 count of 256-bit units							
		(See Stream 0 Vertex Read Length)							
	23:22	Reserved							
		Format:	MBZ						
	21	Stream 2 Vertex Read Offset							
		Format: U1 count of 256-bit units							
		Specifies amount of data to skip over before reading back Stream 2 vertex data. (See Stream 0 Vertex Read Offset)							
	20:16	Stream 2 Vertex Read Length							
		Format: U5-1 count of 256-bit units							
	15:14	Reserved							
		Format:	MBZ						
	13	Stream 1 Vertex Read Offset							
		Format: U1 count of 256-bit units							
		Specifies amount of data to skip over before reading back S Vertex Read Offset)	Stream 1 vertex data. (See Stream 0						
	12:8	Stream 1 Vertex Read Length							



	3	DSTATE_STREAMOUT						
	Format:	U5-1 count of 256-bit units						
	(See Stream 0 Vertex Read Length)							
7:6	Reserved							
	Format:	MBZ						
5	Stream 0 Vertex Rea	Stream 0 Vertex Read Offset						
	Format:	U1 count of 256-bit units						
	the GS is enabled and the Output Vertex Size field in 3DSTATE_GS is programmed to 0 (i.e., one 16B unit).							
4:0	Stream 0 Vertex Rea	ad Length						
	Format:	U5-1 count of 256-bit units						
	Specifies amount of vertex data to read back for Stream 0 vertices, starting at the Stream 0 Vertex Read Offset location. Maximum readback is 17 256-bit units (34 128-bit vertex attributes). Read data past the end of the valid vertex data has undefined contents, and therefore shouldn't be used to source stream out data. Must be zero (i.e., read length = 256b) if the GS is enabled and the Output Vertex Size field in 3DSTATE_GS is programmed to 0 (i.e., one 16B unit).							



	3DSTATE_TE										
Source:		R	lenderCS								
Length B	ias:	2									
The state	e used	by TE is	defined with this inlir	oacket.							
DWord	Bit	Description									
0	31:29	Comma	and Type								
		Defaul	t Value:				3h G	FXPIPE			
		Format	t:				ОрСо	ode			
	28:27	Comma	and SubType								1
		Defaul	t Value:		3h GF	XPIPE	E_3D				
		Format:					de				
	26:24	3D Con	nmand Opcode		1						
		Defaul	t Value:	0h 3DS	TATE_PI	PELIN	IED				
		Format	t:	OpCod	е						
	23:16	3D Con	nmand Sub Opcode								
		Defaul	t Value:		1Ch 3DSTATE_TE						
		Format	t:			OpCode	9				
	15:8	Reserved									
		Format	t:					MBZ			
	7:0	DWord	l Length	i							
		Defaul	t Value:		2h Exclu	ides DW	ord (0	0,1)			
		Format	t:		=n Tota	I Length	- 2				
1	31:19	Reserve	ed								
		Format	t:					MBZ			
	18:16	Reserve	ed]
		IFORMAT: MBZ									
	15:14	Reserved									
	1212	Format	•					IVIBZ			
	13:12	Partitic	oning						112]
		This fiel	ι. Id specifies how edge	es are par	titioned	based o	on tes	sellation	102 factor		
		Value	Name				Descri	iption	- lucton]
		0h	INTEGER	Outside,	/inside e	edges ar	e divio	ded into	an integer	number	_
			of equal-sized seg								
		1h	ODD_FRACTIONAL	/inside e -unequa	edges ar al-sized	e divio segmo	ded into ents.	an odd nu	mber of		
		2h EVEN_FRACTIONAL Outside/inside edges are divided into an						an even nu	umber of		
	11:10	Reserve	ed								



				3DSTATE_TE					
	Format	t:			MBZ				
9:8	Output Topology								
	Format: U2								
	This field specifies which primitive types are to be output.								
	Value	Name		Description	n		_		
	0h	POINT	Points	s are output (as POINTLIST topolog	gies)				
	1h	LINE	Lines doma	ines are output (as LINESTRIP topologies). Only valid if ISOLINE Iomain is selected.					
	2h	TRI_CW	Clock TRIST select	ockwise-ordered triangles are output (either as TRISTRIP, RISTRIP_REV or TRILIST topologies). Not valid if ISOLINE domain is elected.					
	3h	TRI_CCW	Count TRIST select	Count-clockwise-ordered triangles are output (either as TRISTRIP, 'RISTRIP_REV or TRILIST topologies). Not valid if ISOLINE domain is elected.					
7:6	Reserve	ed							
	Format	t:			MBZ				
5:4	TE Don	nain							
	Format	t:				U2			
	This fiel	This field specifies which type of domain is to be tessellated.							
	Value Name			Description	on				
	0h	QUAD		2D (U,V) domain is tessellated					
	1h	TRI		Triangular (U,V,W) domain is tessellated					
	2h	ISOLIN	E	2D (U,V) domain is tessellated.					
3	Reserve	ed							
	Format	t:			MBZ				
2:1	TE Moo	de							
	Format	t:				U2			
	When T	E Enable is	ENAB	BLED, this field specifies the overall	operatio	on of the TE stage. Thi	s field is		
	ignored	l if TE Enab	le is D	ISABLED.			1		
	Value	Name		Descriptio	n				
	Uh	HW_IESS	Norm patch tessel	nal HW Tessellation Mode. The Tes I URB entry, and are used to perfor Ilation of the specified domain.	sFactors rm fixed	are read from the -function hardware			
	lh	SW_TESS	Software Tessellation Mode. The TE unit will pass down HS-thread- generated tessellated domain points instead of generating them itself from TessFactors. The TE unit will read the Domain Point Count and Domain Point Buffer Starting Address fields from the patch header, and if the count is 0 it will consider the patch culled and discard it. Otherwise the address is used to start fetching DOMAIN_POINT structures from memory and passing them down the pipeline to DS.						



	3DSTATE_TE									
		2h Reserved Reserved								
		3h Reserved Reserved								
	0	TE Enal	ble				1			
		Forma	t:			Enable				
		If ENABLED, the TE stage will perform tessellation processing on incoming patch primitives. The TE Mode field determines how this tessellation operation proceeds. If DISABLED, the TE goes into pass-through mode. All other state fields are ignored.								
						Programming Notes				
		The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.								
2	31:0	Maxim	um Tessel	lation F	actor Odd	d				
		Forma	t:			IEEE_Float				
		This fiel HW_TE	This field specifies the maximum TessFactor for ODD_FRACTIONAL partitioning when in HW TESS mode.							
			Value		Name	Description				
		427c00	000h 63 Per API Spec, For normal operation software should set this value to 63.0							
		[40400	000h,427c	0000h]	Reserved	Reserved.				
		Programming Notes								
		Note t the Pa	hat ISOLIN rtitioning s	E's Linel tate.	Density TF	is always subjected to INTEGER partitioning regardless	s of			
3	31:0	Maxim	um Tessel	lation F	actor Not	t Odd				
		Forma	t:			IEEE_Float				
		This fiel when ir	ld specifies	the ma mode.	ximum Te	ssFactor for EVEN_FRACTIONAL or INTEGER partitioning	g			
			Value		Name	Description				
	42800000h 64 Per API Spec, For normal operation software should value to 64.0						et this			
		[4000000h,42800000h] Reserved Reserved								
		Netet			Donaity	rogramming Notes	. of			
		the Pa	rtitioning st	tate.	Density IF	is always subjected to intreGER partitioning regardless	5 01			



		3DST/		_UR	B_C	DS		
Source:	Source: RenderCS							
Length Bi	Length Bias: 2							
This command may not overlap with the push constants in the URB defined by the 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, and 3DSTATE_PUSH_CONSTANT_ALLOC_GS commands.						by the _DS, LOC_GS commands.		
		Prog	ırammi	ing No	otes			
3DSTATE program	URB_V ming of	'S, 3DSTATE_URB_HS, and 3DSTAT f this state to be valid.	re_urb	8_GS m	ust also	o be pr	ogrammed in order for the	
DWord	Bit	Description						
0	31:29	Command Type Default Value:				3h GFX		
	20.27					ορεοι		
	28:27	Default Value			3h GE			
		Format:				de		
	26.24	3D Command Opcode		I	<u> </u>			
	20.21	Default Value:	0h	3DSTA	ATE_PI	PELINEI	D	
		Format: OpCode						
	23:16	3D Command Sub Opcode	<u> </u>					
		Default Value:	32	2h 3DS	3DSTATE_URB_DS			
		Format:	0	DpCode	9			
	15:8	Reserved						
		Format:				Ν	1BZ	
	7:0	DWord Length						
		Default Value:	h DWC	ORD_C	DUNT_	n		
		Format:	=	n				
1	31	Reserved					107	
		Format:				IV	IBZ	
	30	Reserved					107	
	20.25					IV	IDZ	
	29:25	DS URB Starting Address	DS URB Starting Address					
	nemory	y wher	e DS st	tarts its	allocation, specified in multiples of			
		8 KB.					1	
		Value		Nam	e			
		[0,11]						
	24:16	DS URB Entry Allocation Size		10.1.1	•.			
		Format: U9-1 Cour	nt of 51	12-bit	units	Thicf	ield is always used (aven if DS	
		specifies the length of each ORB	entry	owned	DY DS	. THIS TI	ieiu is always useu (even n DS	



	3DSTATE_URB_DS									
		Function Enable is DISABLED).								
		Value			Name					
		[0,9]								
1	.5:0	DS Number of URB Entries								
			Descripti	on						
		Specifies the number of URB entries that are used by DS. This field is always used (even if DS Function Enable is DISABLED).								
		If Domain Shader Thread Dispatch is Enabled then the minimum number of handles that must be allocated is 10 URB entries.								
		Value	Nai	me]					
		[0,288]								
		Des unservices Nistes								
			Program	ming Notes						
		DS Number of URB Entries must be divisible by 8 if the DS URB Entry Allocation Size is programmed to a value less than 9, which is 10 512-bit URB entries. "2:0" = reserved "000								



		3D9	ТАТ	E_UF	RB_(GS			
Source:		RenderCS							
Length E	Length Bias: 2								
This command may not overlap with the push constants in the URB defined by the 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, and 3DSTATE_PUSH_CONSTANT_ALLOC_GS commands.						d by the)C_DS, ALLOC_GS commands.			
			Program	nming N	otes				
3DSTAT prograr	TE_URB_ mming	_VS, 3DSTATE_URB_HS, and 3D of this state to be valid.	STATE_L	JRB_DS n	nust als	so be	programmed in order for the		
DWord	Bit			Dese	riptio	n			
0	31:29	Command Type							
		Default Value:				3h G	FXPIPE		
		Format:				ОрСо	ode		
	28:27	Command SubType							
		Default Value:				XPIPE	_3D		
		Format:				de			
	26:24	3D Command Opcode							
		Default Value:	(0h 3DSTA	h 3DSTATE_PIPELINED				
		Format: OpCode							
	23:16	3D Command Sub Opcode		•					
		Default Value:		33h 3DS	STATE_	URB_	GS		
		Format:		OpCode	•				
	15:8	Reserved					MRZ		
	7.0						IVIDZ		
	7:0	Default Value:					[]		
		Default Value: Oh DW							
	21			-11					
T	31	Keserved					MPZ		
	20								
	30	Keserved Format:					MRZ		
	20.25						IVIDZ		
	29:25	GS URB Starting Address							
		Offset from the start of the LIR	R memo	ary where	GS st	arts it	s allocation specified in multiples of 8		
		KB.	Offset from the start of the URB memory where GS starts its allocation, specified in multiples of 8 KB.						
		Value		Nam	e				
		[0,11]							
	24:16	GS URB Entry Allocation Size	;						
		Format:	U9-1 5	12-bit un	its				
		Specifies the length of each U	RB entrv	v owned k	ov GS. [·]	This f	ield is always used (even if GS		



	3DS ⁻	TATE_URB_GS				
	Function Enable is DISABLED).					
15:0 GS Number of URB Entries Specifies the number of URB entries that are used by GS. This field is always used (even if Function Enable is DISABLED).						
	Value	Name				
	[0,192]					
		Programming Notes				
	Only if GS is disabled can this field be programmed to 0. If GS is enabled this field shall be programmed to a value greater than 0. For GS Dispatch Mode "Single", this field shall be programmed to a value greater than or equal to 1. For other GS Dispatch Modes, refer to the definition of Dispatch Mode (3DSTATE_GS) for minimum values of this field.					
	GS Number of URB Entries mus 9 512-bit URB entries. "2:0" = reserved "000"	t be divisible by 8 if the GS UF	B Entry Allocation Size is less than			



		3DST		E_UR	B_H	IS		
Source:	Source: RenderCS							
Length Bi	Length Bias: 2							
This command may not overlap with the push constants in the URB defined by the 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, and 3DSTATE_PUSH_CONSTANT_ALLOC_GS commands.								
		Prog	gram	ming No	otes			
3DSTATE_URB_VS, 3DSTATE_URB_DS, and 3DSTATE_URB_GS must also be programmed in order for the programming of this state to be valid.						ogrammed in order for the		
DWord	Bit			Dese	riptio	n		
0	31:29	Command Type						
		Default Value:				3h GFX	PIPE	
		Format:				OpCod	e	
	28:27	Command SubType						
		Default Value:			3h GF	XPIPE_3	3D	
		Format:		OpCo	de			
	26:24	3D Command Opcode						
		Default Value:	()h 3DSTA	1 3DSTATE_PIPELINED			
		Format: OpCode						
	23:16	3D Command Sub Opcode						
		Default Value:		31h 3D9	1h 3DSTATE_URB_HS			
		Format:		OpCode	9			
	15:8	Reserved						
		Format:				Μ	IBZ	
	7:0	DWord Length						
		Default Value:	0h DWC	ORD_C	JUNT_I	n		
		Format:		=n				
1	31	Reserved						
		Format:				Μ	IBZ	
	30	Reserved						
		Format:				Μ	IBZ	
29:25 HS URB Starting Address								
		Format: U5						
Offset from the start of the URB memory where HS starts its					allocation, specified in multiples of			
		Value		Nam	e]	
		[0,11]						
	24:16	HS URB Entry Allocation Size						
		Format: U9-1 Cou	int of	512-bit	units			
		Specifies the length of each URE	8 entr	y owned	by HS	. This fi	eld is always used (even if HS	



3DSTATE_URB_HS								
	Function Enable is DISABLED)							
15:0	HS Number of URB Entries Specifies the number of UR if HS Function Enable is DIS Programming Restriction:H URB Entry Allocation Size is Value [0,32]	RB entries that are used by H SABLED). IS Number of URB Entries m s less than 9 512-bit URB en Name	IS. This field is always used (even ust be divisible by 8 if the HS tries."2:0" = reserved "000"					

RenderCS



3DSTATI	E URB VS

Source:

Length Bias: 2

Description

VS URB Entry Allocation Size equal to 4(5 512-bit URB rows) may cause performance to decrease due to banking in the URB. Element sizes of 16 to 20 should be programmed with six 512-bit URB rows.

This command may not overlap with the push constants in the URB defined by the 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, and 3DSTATE_PUSH_CONSTANT_ALLOC_GS commands.

Programming Notes

3DSTATE_URB_HS, 3DSTATE_URB_DS, and 3DSTATE_URB_GS must also be programmed in order for the programming of this state to be valid.

DWord	Bit	Description					
0	31:29	Command Type					
		Default Value:			3h GFXPIPE		
		Format:		OpCo	ode		
	28:27	Command SubType					
		Default Value:		3h GFXPIPE	_3D		
		Format:		OpCode			
	26:24	3D Command Opcode	_				
		Default Value:	0h 3DST	ATE_PIPELIN	ED		
		Format:	OpCode				
	23:16	3D Command Sub Opcode	3D Command Sub Opcode				
		Default Value:	30h 3D	30h 3DSTATE_URB_VS			
		Format:	Format: OpCode				
	15:8	Reserved					
		Format: MBZ					
	7:0	DWord Length					
		Default Value:	0h DW	ORD_COUNT	_n		
		Format:	=n				
1	31	Reserved					
		Format:			MBZ		
	30	Reserved					
		Format: MBZ					
	29:25	VS URB Starting Address					
		Format:			U5		
		Offset from the start of the URB mer	nory whe	re VS starts it	s allocation, specified in multiples of		
		ö KB.	Nor	20			
		Value	INdi				



	3DSTATE_URB_VS								
	[0,11]								
24:	16 VS URB Entry Allocat	VS URB Entry Allocation Size							
	Format:	U9-1 count of 512	2-bit units						
Specifies the length of each URB entry owned by VS. This field is always used (even if VS Function Enable is DISABLED).									
		Prog	gramming Notes						
Programming Restriction: As the VS URB entry serves as both the per-vertex input an of the VS shader, the VS URB Allocation Size must be sized to the maximum of the vertice and output structures.									
15	0 VS Number of URB E	ntries							
	Format:			U16					
	Specifies the number Function Enable is DIS	of URB entries that ABLED).	are used by VS. This	s field is always used (even if VS					
	Value		Name						
	[32,512]								
Programming Notes									
	Programming Restrict Allocation Size is less	ion: VS Number of than 9 512-bit URB	URB Entries must be entries."2:0" = rese	e divisible by 8 if the VS URB Entry rved "000b"					



3DSTATE_VERTEX_BUFFERS

Source:

RenderCS

Length Bias: 2

Description

This command is used to specify VB state used by the VF function.

Can specify from 1 to 33 VBs.

The VertexBufferID field within a VERTEX_BUFFER_STATE structure indicates the specific VB. If a VB definition is not included in this command, its associated state is left unchanged and is available for use if previously defined.

Programming Notes

It is possible to have individual vertex elements sourced completely from generated ID values and therefore not require any vertex buffer accesses for that vertex element. In this case, VF function will simply ignore the VB state associated with that vertex element. If all enabled vertex elements have this characteristic, no VBs are required to process 3DPRIMITIVE commands. For example, this might arise when the user wants to perform all data lookups in the first shader, so only generated index values need to be passed down to it. In this extreme case, SW would not need to program any VB state, and therefore not need to issue any 3DSTATE_VERTEX_BUFFERS commands.

For any 3DSTATE_VERTEX_BUFFERS command, at least one VERTEX_BUFFER_STATE structure must be included.

VERTEX_BUFFER_STATE structures are 4 DWords for both VERTEXDATA buffers and INSTANCEDATA buffers.

Inclusion of partial VERTEX_BUFFER_STATE structures is UNDEFINED.

The order in which VBs are defined within this command can be arbitrary, though a vertex buffer must be defined only once in any given command (otherwise operation is UNDEFINED).

DWord	Bit	Description					
0	31:29	Command Type					
		Default Value:			03h GFX	(PIPE	
		Format:			Opcode	,	
	28:27	Command SubType					
		Default Value:				3h 3D	
		Format:				Opcode	
	26:24	3D Command Opcode					
		Default Value:	0h 3DS	TATE_VE	RTEX_BL	UFFERS	
		Format:	Opcod	е			
	23:16	3D Command Sub Opcod	le				
		Default Value:	08h 3D9	STATE_VE	RTEX_BU	BUFFERS	
		Format:	Opcode	9			
	15:8	Reserved					
	7:0	DWord Count					
		Default Value: 3 DWOR		D_COUN	T_n		
		Format:		=n			
		n = 4b-1 (where $b = #$ of b	ouffer sta	tes inclu	ded)		



3DSTATE_VERTEX_BUFFERS						
1n	127:0	Vertex Buffer State [n]				
		Format:	VERTEX_BUFFER_STATE			



3DSTATE	VERTEX	ELEMENTS
	-	

Source:

RenderCS

Length Bias: 2

Description

This is a variable-length command used to specify the active vertex elements. Each VERTEX_ELEMENT_STATE structure contains a Valid bit which determines which elements are used.

Up to 34 elements.

Programming Notes

At least one VERTEX_ELEMENT_STATE structure must be included.

Inclusion of partial VERTEX_ELEMENT_STATE structures is UNDEFINED.

SW must ensure that at least one vertex element is defined prior to issuing a 3DPRIMTIVE command, or operation is UNDEFINED.

There are no 'holes' allowed in the destination vertex: NOSTORE components must be overwritten by subsequent components unless they are the trailing DWords of the vertex. Software must explicitly chose some value (probably 0) to be written into DWords that would otherwise be 'holes'.

Within a VERTEX_ELEMENT_STATE structure, if a Component Control field is set to something other than VFCOMP_STORE_SRC, no higher-numbered Component Control fields may be set to VFCOMP_STORE_SRC. In other words, only trailing components can be set to something other than VFCOMP_STORE_SRC.

See additional restrictions listed in the command fields and VERTEX_ELEMENT_STATE description.

Element[0] must be valid.

All elements must be valid from Element[0] to the last valid element. (I.e. if Element[2] is valid then Element[1] and Element[0] must also be valid).

The pitch between elements packed in the URB will always be 128 bits.

DWord	Bit		Description				
0	31:29	Command Type					
		Default Value:	(03h GFX	(PIPE		
		Format:	(Opcode			
	28:27	Command SubType					
		Default Value:			3h 3D		
		Format:			Opcode		
	26:24	3D Command Opcode					
		Default Value:	0h 3DSTATE_VERT	EX_ELEN	/ENTS		
		Format: Opcode					
	23:16	3D Command Sub Opcod	e		_		
		Default Value:	09h 3DSTATE_VERT	EX_ELE	MENTS		



3DSTATE_VERTEX_ELEMENTS									
		Format:		Opcode					
	15:8	Reserved	Reserved						
	7:0	DWord C)Word Count						
		Format:				=n			
		Vertex Ele	ement Count	= (DWord Count + 1) / 2	2				
		Value		Name	Description				
		1	DWORD_CC)UNT_n [Default]	excludes D	Words 0,1			
		[1,66]	Range		1-34 Elem	ents			
1n	63:0	Element [n]							
		Format:		VERTEX_ELEMENT_STAT	E				



3DSTATE_VF_STATISTICS

Source:		Render	CS						
Length B	ias:	1							
The VF st VF will in 3DSTATE	age tra cremen _VF_ST	cks two pipeli It the appropr ATISTICS com	ne statistics, the nu iate counter for eac mand with the [Stat	mbe h wh: tistic:	r of vertices fetc nen statistics gat s Enable] bit set.	ched a thering	and the number of g is enabled by issu	objects generated. Jing the	
DWord	Bit				Descriptio	on			
0	31:29	Command T	уре						
		Default Valu	e:			3h GFXPIPE			
		Format:				Орсо	ode		
	28:27	Command S	ubType						
Format: Opcode									
1h Pipelined Single DWord [Default]									
	26.24	2D Common	In Pipelined, Single Dword [Default]						
	26:24	Dofault Valu	BD Command Opcode						
		Format:	Oncode	pcode					
		GFXPIPE[28:2	7 = 1h. 26:24 = 0h.	23:1	6 = 0Bh1 (Pipelir	ined. Single DWord)			
			, ,		1 (, -	<u> </u>		
	23:16	3D Comman	d Sub Opcode						
		Default Valu	e:	0Bh	h 3DSTATE_VF_STATISTICS				
		Format:		Орс	ode				
		GFXPIPE[28:2	7 = 1h, 26:24 = 0h,	23:1	6 = 0Bh] (Pipelir	ned, S	ingle DWord)		
	15:1	Reserved							
		Format:					MBZ		
	0	Statistics Ena	able						
		Format:			E	Inable			
		If ENABLED, \	/F will increment th	e pip	peline statistics o	counte	ers IA_VERTICES_CC	OUNT and	
			S_COUNT for each	verte	ex fetched and e	each o	bject output, respe	ctively, for	
		If DISABLED.	these counters will	not	be incremented	for su	ubsequent 3DPRIM	ITIVE commands.	
		-,						,	

٦



		3DSTATE_\	/IEWPOR	T_ST	ATE_P	OINTERS_CC			
Source:		RenderCS							
Length B	Bias:	2							
The 3DS viewport	STATE_ t state f	VIEWPORT_STATE_PC table.	DINTERS_CC com	mand is u	sed to defir	ne the location of fixed functions'			
DWord	Bit			Des	cription				
0	31:29	Command Type			I				
		Default Value:			3h G	FXPIPE			
		Format:			OpC	ode			
	28:27	Command SubType							
		Default Value:			3h GFXPIP	E_3D			
		Format:			OpCode				
	26:24	3D Command Opco	D Command Opcode						
		Default Value:	0h 3DSTATE_PIPELINED						
		Format:		OpCode					
	23:16	3D Command Sub	Opcode						
		Default Value:	23h 3DSTA1	ATE_VIEWPORT_STATE_POINTERS					
		Format:	OpCode						
	15:8	Reserved	Reserved						
		Format:				MBZ			
	7:0	DWord Length							
		Default Value:		0h DWORD_COUNT_n					
		Format:		=n					
1	31:5	CC Viewport Pointe	er						
		Format: Dyr	namicStateOffset	[31:5]CC_\	/IEWPORT*	16			
		Specifies the 32-byte aligned address offset of the CC_VIEWPORT state. This offset is relative to the Dvnamic State Base Address.							
		-							
	4:0	Reserved				[]			
		Format:				MBZ			



	3 C	STATE_VIE	WPORT_	STAT	E_POI	NTERS_SF_CLIP			
Source: Length B	Bias:	RenderCS 2							
The 3DS viewport	STATE_' t state t	VIEWPORT_STATE_PC table.	DINTERS_CLIP cor	nmand is	used to de	fine the location of fixed functions'			
DWord	Bit			Des	cription				
0	31:29	Command Type							
	Default Value:			3h (GFXPIPE				
		Format:			ОрС	Code			
	28:27	Command SubType	•						
		Default Value:		3h GFXPIP	E_3D				
		Format:			OpCode				
	26:24	D Command Opcode							
		Default Value:	Default Value: 0			0h 3DSTATE_PIPELINED			
		Format:		OpCode)pCode				
	23:16	3D Command Sub	Opcode						
		Default Value:	21h 3DSTATE_\	VIEWPOR	EWPORT_STATE_POINTERS_SF_CLIP				
		Format:	OpCode						
	15:8	Reserved							
		Format:				MBZ			
	7:0	DWord Length		L.					
		Default Value:		0h DW0	ORD_COUN	T_n			
		Format:		=n					
1	31:6	SF Clip Viewport Po	ointer						
		Format: DynamicStateOffset[31:6]SF_CLIP_VIEWPORT*16							
		Specifies the 64-byte aligned address offset of the SF_CLIP_VIEWPORT state. This offset is relative to the Dynamic State Base Address.							
	5:0	Reserved							
		Format:				MBZ			



				3DS	ΤΑΤ	E_VS			
Source:		Rei	nderCS						
Length E	Bias:	2							
				Descripti	on				
The stat	te used	by VS is o	defined with	n this inline state	packet.				
DWord	Bit				D	escriptio	on		
0	31:29	Comman	nd Type				1		
		Default \	Value:				3h G	FXPIPE	
		Format:		OpCode					
	28:27	Comman	mmand SubType						
		Default Value:				3h G	FXPIP	E_3D	
		Format:				ОрСо	ode		
	26:24	3D Com	O Command Opcode						
		Default Value: 0h				STATE_P	[PELIN	IED	
		Format:			ОрСос	le			
	23:16	3D Com	D Command Sub Opcode						
		Default	Value:			10h 3D	L0h 3DSTATE_VS		
		Format:				OpCod	е		
	15:8	Reserved	1						
		Format:						MBZ	
	7:0	DWord L	.ength						
		Default \	Value:		4h Exclu	ides DWord (0,1)			
		Format:			=n lota	al Lengtr	1 - 2		
1	31:6	Kernel St	tart Pointe	r Transformer Dana d	266452	1.01/	-1		
		Format:	specifies th		DTISET[3	EN4 co	ei ro inc	truction) of the kernel pro	ogram run by
		threads s	pawned by	this FF unit. It is	specifie	d as a 6^{4}	4-byte	e-granular offset from the	e Instruction
		Base Add	lress. This fi	eld is ignored if \	VS Func	tion Ena	ble is	DISABLED.	
	F 0								
	5:0	Reserved	1					MB7	
n	21	Single V	ortox Diana						
Z	21	Format:	ertex Dispa	U1 Enur	nerated	type			
		This field	can be use	d to force single	vertex S	SIMD4x2	2 VS tł	nreads.	
		Value Name Description							
		0h	0h Multiple Dual vertex SIMD4x2 thread dispatches are allowed.						
		1h	Single	Single vertex SI	MD4x2	thread o	dispat	ches are forced.	
	30	Vector N	lask Enable	e (VME)					
		When SP SPF=1, V	F=0, VME s ME specifie	pecifies which mask to	ask to u use to c	ise to ini generate	tialize exect	e the initial channel enable ution channel enables.	es. When



	Value	Name		De	scriptio	n			
	0h	Dmask	Channels are	enabled based of	on the d	ispatch ma	isk		
	1h	Vmask	Channels are	enabled based o	on the v	ector mask			
29:27	Sampler Specifies prefetchir DISABLED	Count how many ng the asso	samplers (in n ociated sample	nultiples of 4) th r state entries. T	ie vertex This field	shader 0 k is ignored	kernel uses if VS Func	. Used onl tion Enabl	y f le i
	Value		Name		Des	cription			
	0h	No Samplers		no samplers us	sed				
	1h	1-4 Sam	plers	between 1 and	l 4 samp	lers used			
	2h	5-8 Sam	plers	between 5 and	l 8 samp	lers used			
	3h	9-12 Sar	nplers	between 9 and	l 12 sam	plers used			
	4h 13-16 Samplers			between 13 an	d 16 sar	mplers used	d		
26	Reserved Format:	Reserved Format: MBZ							
25:18	Binding Table Entry Count								
	Format: U8								
	binding ta	able entrie kernels us	s and associate	entries the kern ed surface state. mber of binding	el uses. table el	ntries, it ma	for prefetc ay be wise	hing of th to set this	e fie
	binding ta Note: For zero to av This field	able entrie kernels us oid prefet is ignored	s and associate ing a large nui ching too man if VS Function Value	entries the kern ed surface state. mber of binding y entries and th Enable is DISAB	table en rashing LED.	ntries, it ma the state c	for prefetc ay be wise ache. Name	hing of th to set this	e fi
	binding ta Note: For zero to av This field	able entrie kernels us oid prefet is ignored	s and associate ing a large nui ching too man if VS Function Value	entries the kern ed surface state. mber of binding ly entries and th Enable is DISAB	table en rashing LED.	ntries, it mathematical the state c	for prefetc ay be wise ache. Name	hing of th	e fi
17	binding ta Note: For zero to av This field [0,255] Reserved	able entrie kernels us void prefet is ignored	s and associate ing a large nu ching too man if VS Function Value	entries the kern ed surface state. mber of binding iy entries and th Enable is DISAB	table en rashing BLED.	ntries, it ma the state c	for prefetc ay be wise ache. Name	hing of th	e fi
17	binding ta Note: For zero to av This field [0,255] Reserved Format:	able entrie kernels us void prefet is ignored	s and associate ing a large nu ching too man if VS Function Value	entries the kern ed surface state. mber of binding by entries and th Enable is DISAB	el uses. table el rashing BLED.	ntries, it ma the state c	for prefetc ay be wise ache. Name	hing of th	e fi
17	binding ta Note: For zero to av This field [0,255] Reserved Format: Floating	Point Mod	s and associate ing a large nui ching too man if VS Function Value	entries the kern ed surface state. mber of binding y entries and th Enable is DISAB	el uses. table en rashing BLED.	MBZ	for prefetc ay be wise ache. Name	hing of th	e fie
17 16	binding ta Note: For zero to av This field [0,255] Reserved Format: Floating Format:	Point Mod	s and associate ing a large nui ching too man if VS Function Value de	entries the kern ed surface state. mber of binding ny entries and th Enable is DISAB	el uses. table en rashing BLED.	MBZ	for prefetc ay be wise ache. Name	hing of th	e fie
17 16	binding ta Note: For zero to av This field [0,255] Reserved Format: Floating Format: Specifies	Point Mod the initial t	s and associate ing a large nui ching too man if VS Function Value de U1 er floating point r	entries the kern ed surface state. mber of binding iy entries and th Enable is DISAB	el uses. table en rashing BLED.	MBZ	for prefetc ay be wise ache. Name ad. This fiel	hing of the to set this	e fie
17	binding ta Note: For zero to av This field [0,255] Reserved Format: Floating Format: Specifies Function	Point Mod the initial t Enable is E	s and associate ing a large nui ching too man if VS Function Value de U1 er floating point r DISABLED. Name	entries the kern ed surface state. mber of binding by entries and th Enable is DISAB	table en rashing BLED.	MBZ	for prefetc ay be wise ache. Name ad. This fiel	hing of th to set this d is ignore	e fie
17 16	binding ta Note: For zero to av This field [0,255] Reserved Format: Floating Format: Specifies Function Valu Oh	Point Moo the initial t Enable is E	s and associate ing a large nui ching too man if VS Function Value de U1 er floating point r DISABLED. Name	entries the kern ed surface state. mber of binding ay entries and th Enable is DISAB numerated type mode used by th Use IEEE-754	table en rashing BLED. ne dispa Descr 4 Rules	MBZ	for prefetc ay be wise ache. Name ad. This fiel	hing of th to set this d is ignore	e fie ed
17	binding ta Note: For zero to av This field [0,255] Reserved Format: Floating Format: Specifies Function Oh 1h	Point Mod the initial t Enable is E Alte	s and associate ing a large nui ching too man if VS Function Value de U1 er floating point r DISABLED. Name :-754	entries the kern ed surface state. mber of binding ay entries and th Enable is DISAB numerated type mode used by th Use IEEE-754 Use alternat	table en rashing BLED. ne dispa Descr 4 Rules e rules	MBZ	for prefetc ay be wise ache. Name	hing of the to set this	e fit
17 16 15:14	binding ta Note: For zero to av This field [0,255] Reserved Format: Floating Format: Specifies Function Valu Oh 1h Reserved	Point Moo the initial t Enable is C e IEEE	s and associate ing a large nui ching too man if VS Function Value de U1 er floating point r DISABLED. Name E-754 ernate	entries the kern ed surface state. mber of binding ay entries and th Enable is DISAB numerated type mode used by th Use IEEE-754 Use alternat	table en rashing BLED. ne dispa Descr 4 Rules e rules	MBZ	for prefetc ay be wise ache. Name ad. This fiel	hing of th to set this d is ignore	e fie ed
17 16 15:14	binding ta Note: For zero to av This field [0,255] Reserved Format: Floating Format: Specifies Function Oh 1h Reserved Format:	Point Moo the initial t Enable is E Alte	de Ul er floating point r DISABLED. Name	entries the kern ed surface state. mber of binding ay entries and th Enable is DISAB numerated type mode used by th Use IEEE-754 Use alternat	el uses. table en rashing BLED. he dispa Descr 4 Rules e rules	MBZ	for prefetc ay be wise ache. Name ad. This fiel	hing of the to set this	e fie ed
17 16 15:14 13	binding ta Note: For zero to av This field [0,255] Reserved Format: Floating Format: Specifies Function U 0h 1h Reserved Format: Illegal Op	Point Moo the initial f Enable is E e IEEE Alte	s and associate ing a large nui ching too man if VS Function Value de U1 er floating point r DISABLED. Name E-754 ernate	entries the kern ed surface state. mber of binding ay entries and th Enable is DISAB numerated type mode used by th Use IEEE-754 Use alternat	table en rashing BLED. ne dispa Descr 4 Rules e rules	MBZ	for prefetc ay be wise ache. Name ad. This fiel	hing of the to set this	e fie ed



			3	DSTATE_VS						
	12	Reserved								
		Format:			MBZ					
	11:8	Reserved								
		Format:			MBZ					
	7	Software Exce	ption Enable							
		Format:		Enable	9					
		This bit gets loa Execution Envir	aded into EU CR onment.This fiel	0.1[13] (note the bit # differed is ignored if VS Function	rence). S Enable is	ee Exceptions and ISA s DISABLED.				
	6:0	Reserved	Reserved							
	010	Format:	Format: MBZ							
3	31:10	Scratch Space	Base Offset							
5	51.10	Format:	GeneralState	eOffset[31:10]ScratchSpace						
		aligned offset fi unit will be allo computed offse Space Offset. Th scratch space, v offset passed in This field is igno	pecifies the starting location of the scratch space area allocated to this FF unit as a 1K-byte iligned offset from the General State Base Address. If required, each thread spawned by this FF init will be allocated some portion of this space, as specified by Per-Thread Scratch Space. The computed offset of the thread-specific portion will be passed in the thread payload as Scratch Space Offset. The thread is expected to utilize "stateless" DataPort read/write requests to access scratch space, where the DataPort will cause the General State Base Address to be added to the offset passed in the request header. This field is ignored if VS Function Enable is DISABLED.							
	9:4	Reserved								
		Format:			MBZ					
	3:0	Per-Thread Sci	ratch Space							
		Format:	U4 power	of 2 Bytes over 1K Bytes						
		Specifies the an The driver must Pointer, to ensu size without exc Enable is DISAB	nount of scratch allocate enoug ire that the Max ceeding the driv BED.	n space to be allocated to each h contiguous scratch space imum Number of Threads of er-allocated scratch space.	ach threa , starting can each This field	ad spawned by this FF unit. g at the Scratch Space Base get Per-Thread Scratch Space d is ignored if VS Function				
		Value	Name		Descrip	tion				
		[0,11]		indicating [1K Bytes, 2M By	ytes]					
				Programming Note	S					
		This amount is available to the kernel for information only. It will be passed verbatim (if not altered by the kernel) to the Data Port in any scratch space access messages, but the Data Port will ignore it.								
4	31:25	Reserved								
		Format:			MBZ					
	24:20	Dispatch GRF S	Start Register f	or URB Data		115				
		i offiat.				05				



				3DS	TATE_	VS				
		Specifies the st thread payload	tarting d. This	g GRF register nu field is ignored	umber for th if VS Functi	ne URB poi on Enable	rtion (Constant + Vert is DISABLED.	ices) of the		
		Value		Name			Description			
		[0,31]			indicating	GRF [R0,R	31]			
	19:17	Reserved								
		Format:					MBZ			
	16:11	Vertex URB E	ntry R	lead Length						
		Format:					U6			
		Specifies the number of pairs of 128-bit vertex elements to be passed into the payload for each vertex. This field is ignored if VS Function Enable is DISABLED.								
		the number of	of GRI	Fs with vertex c	ata will be	double t	be value programm	ed in this field		
							Name			
		[1.63]								
		[_/~~]								
					Program	ning Note	25			
		It is UNDEFIN the thread.	ED to	set this field to () indicating	no Vertex	URB data to be read a	and passed to		
	10	Reserved								
		Format:					MBZ			
	9:4	Vertex URB E	ntry R	lead Offset			l.			
		Format:					U6			
		Specifies the o being included thread. This fie	ffset (I in the Id is ig	in 256-bit units) e thread payloac gnored if VS Fun	at which Ve d. This offse action Enabl	ertex URB o t applies to e is DISAB	data is to be read fron o all Vertex URB entrie LED.	n the URB before as passed to the		
				Value			Name			
		[0,63]								
	3:0	Reserved								
		Format:					MBZ			
5	31:25	Maximum Nu	mber	of Threads						
		Format:		U7-1 represent	ting thread	count				
	Specifies the maximum number of simultaneous threads allowed to be active. Used to avoid using up the scratch space. Programming the value of the max threads over the number of threads based off number of threads supported in the execution units may improve perfor since the architecture allows threads to be buffered between the check for max threads an									
		threads suppo	rted ir	the execution i	uning the m units may re	ax values i duce perfe	ormance. This field is i	anored if VS		
		Function Enabl	le is D	ISABLED.				ghorea in vo		
		Value			Nar	ne				
		[0,15]	indica	ating thread cou	nt of [1,16]					
	24:23	Reserved								



	3DSTATE_V	'S					
	Format:		MBZ				
22:11	Reserved						
	Format:		MBZ				
10	Statistics Enable						
	Format:	Enable	e				
	Description						
	If ENABLED, this FF unit will engage in statistics gathering. See the Statistics Gathering section later in this chapter. If DISABLED, statistics information associated with this FF stage will be left unchanged.						
	This field is used even if VS Function Enable is DISABLED.						
9:3	Reserved		1				
	Format:		MBZ				
2	Reserved						
	Format:		MBZ				
1	Vertex Cache Disable			1			
	Format: Disable						
	is DISABLED and the VS Function is ENABLED, the V vertices will be passed to VS threads. If the Vertex Cache is ENABLED and the VS Functio hit in the Vertex Cache will be passed to VS thread If the Vertex Cache is ENABLED and the VS Functio Vertex Cache will be assembled and written to the unmodified (not shaded). The Vertex Cache is invalidated whenever the Vertev VS Function Enable toggles, between 3DPRIMITIVE 3DPRIMITIVE command.	Vertex C on is EN/ s. on is DIS URB, th ex Cache comma	Cache is not used and all incom ABLED, incoming vertices that of ABLED, input vertices that miss ough pass thru the VS stage e becomes DISABLED , whenev ands and between instances wi	ing do not in the rer the thin a			
0	VS Function Enable						
	Format:	Enable	e				
	Description						
	Description If ENABLED, VS threads may be spawned to process VF-generated vertices before the resulting vertices are passed down the pipeline. If DISABLED, VF-generated vertices will pass thru the VS function and sent down the pipeline unmodified. The Vertex Cache is still available in this mode, if enabled.						
	If Statistics Enable is ENABLED, VS_INVOCATION_overtex that passes through the VS stage, even if V This field is always used.	COUNT S Functi	will increment by 1 for every ion Enable is DISABLED.				



		3DS	TATE	_WN	Л			
Source:		RenderCS						
Length E	Bias:	2						
DWord	Bit		D	escriptio	on			
0	31:29	Command Type						
		Default Value:			3h G	FXPIPE		
		Format:			OpCode			
	28:27	Command SubType						
		Default Value:		3h G	FXPIPE	_3D		
		Format:			ode			
	26:24	3D Command Opcode						
		Default Value: 0h 3DS			IPELIN	IED		
		Format: OpCode						
	23:16	3D Command Sub Opcode						
		Default Value:	1	.4h 3DST	ATE_V	VM		
		Format:	C	DpCode				
	15:8	Reserved						
		Format: MBZ						
	7:0	DWord Length						
		Default Value:	01h Excl	udes DV	Vord ((0,1)		
		Format:	=n					
		Total Length - 2						
1	31	Statistics Enable						
		Format:			Enable	2		
		If ENABLED, the Windower and pixe	el pipeline	e will eng	gage i	n statistics gathering. If DISABLED,		
		Gathering.	n unis ff	stage w	in be i	en unchanged. See Statistics		
	30	Depth Buffer Clear						
		Format:		I	Enable	2		
		When set, the depth buffer is initial	ized as a	side-effe	ect of	rendering pixels.		
			Progra	amming	Note	S		
		If this field is enabled,						
		2. the Depth Test Enable field	d in DEPT	H_STEN	CIL_ST	ATE must be disabled.		
		3. 3DSTATE_DEPTH_BUFFER::D	epth Wri	ite Enabl	e mus [.]	t be set.		
		4. 3DSTATE_DEPTH_BUFFER::S	tencil Wr	ite Enabl	e mus	st be set if		
		3DSTATE_STENCIL_BUFFER:: be set to the correct values.	Stencil b	uffer ena	able is	set. Additionally the following must		



	3DSTATE_VS		
	 DEPTH_STENCIL_STATE::Stencil Write Mask must be 0xFF DEPTH_STENCIL_STATE::Stencil Test Mask must be 0xFF DEPTH_STENCIL_STATE::Back Face Stencil Write Mask must be 0xFF 		
	5. DEPTH_STENCIL_STATE::Back Face Stencil Test Mask must be 0xFF Refer to section 0 "Depth Buffer Clear" for additional restrictions when this field is enabled. If this field is enabled, Pixel Shader Kill Pixel must be disabled.		
29	Thread Dispatch Enable		
	Format:EnableThis bit, if set, indicates that it is possible for a PS thread to modify a render target, i.e., at least one render target is enabled (is not of type SURFTYPE_NULL and has at least one channel enabled for writes) and the PS kernel contains a code path that may issue a write to that/those enabled RTs.		
	Programming Notes		
	This bit is used for performance optimizations and does not directly control writing to render targets. If this bit is DISABLED, no pixel shader threads will be dispatched. For correct behavior, this bit must be set consistently with the behavior of the PS kernel, i.e. if this bit is DISABLED the PS kernel must not write color or depth to any render targets. If this field is disabled, Pixel Shader Kill Pixel must be disabled.		
28	Depth Buffer Resolve Enable		
	Format: Enable		
	When set, the depth buffer is made to be consistent with the hierarchical depth buffer as a side- effect of rendering pixels. This is intended to be used when the depth buffer is to be used as a surface outside of the 3D rendering operation.		
	Programming Notes		
	If this field is enabled,		
	the Depth Buffer Clear and Hierarchical Depth Buffer Resolve Enable fields must both be disabled.		
	3. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set.		
	Refer to section 11.5.4.2 "Depth Buffer Resolve" for additional restrictions when this field is enabled. If Hierarchical Depth Buffer Enable is disabled, enabling this field will have no effect		
27	Hierarchical Depth Buffer Resolve Enable		
	Format: Enable		
	When set, the hierarchical depth buffer is made to be consistent with the depth buffer as a side effect of rendering pixels. This is intended to be used when the depth buffer has been modified outside of the 3D rendering operation.		
	Programming Notes		
	If this field is enabled,		



	30	STATE_VS	
	2. the Depth Buffer Clear a disabled.	nd Depth Buffer Resolve Enable fields must both be	
	3. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set.		
	Refer to section 11.5.4.3 "Hierard when this field is enabled. If Hierarchical Depth Buffer Er Performance Note: expect the H be reduced for some period of the hierarchical depth buffer is initial rendering will tend to bring the H state.	chical Depth Buffer Resolve" for additional restrictions nable is disabled, enabling this field will have no effect. hierarchical depth buffer's impact on performance to ime after this operation is performed, as the lized to a state that makes it ineffective. Further hierarchical depth buffer back to a more effective	
	Software needs to do an ambigu depth buffer width and height an	ate after allocating the surface for the first time if the re NOT aligned to 8 and 4 respectively.	
26	Legacy Diamond Line Rasteriza	tion	
	Format:	Enable	
	Pixel Shader Kill Pixel		
25	Pixel Shader Kill Pixel	s chapter).	
25	Pixel Shader Kill Pixel Format:	Enable	
25	 Pixel Shader Kill Pixel Format: This bit, if ENABLED, indicates (discard) pixels or samples, oth to be ENABLED in the followin The API pixel shader prograthe pixel shader kernel tha received on dispatch. A sampler with chroma key Any render target has Alple The pixel shader kernel ge 	Enable Enable that the PS kernel or color calculator has the ability ther than due to depth or stencil testing. This bit is read g situations: ram contains "killpix" or "discard" instructions, or other cont t can cause the final pixel mask to differ from the pixel material y enabled with kill pixel mode is used by the pixel shader. ha Test Enable or AlphaToCoverage Enable enabled. nerates and outputs oMask.	
25	 Pixel Shader Kill Pixel Format: This bit, if ENABLED, indicates (discard) pixels or samples, oth to be ENABLED in the followin The API pixel shader prograther pixel shader kernel that received on dispatch. A sampler with chroma key Any render target has Alple The pixel shader kernel ge Note: As ClipDistance clipping instructions, there should be made and the pixel shader for the pixel shader kernel de made and the pixel shader for the pixel shader kernel de made and the pixel shader for the pixel shader kernel de made and the pixel shader for the pixel shader kernel de made and the pixel shader for the pixel shader kernel de made and the pixel shader for the pixel shader for the pixel shader for the pixel shader for the pixel shader kernel de made and the pixel shader for the pixel shader for the pixel shader for the pixel shader for the pixel shader kernel de made and the pixel shader for the pixel shader kernel de made and the pixel shader for the pixel shader shader for the pixel shader shader shader shader shader shader shader shader shader shade	Enable that the PS kernel or color calculator has the ability there than due to depth or stencil testing. This bit is rear g situations: ram contains "killpix" or "discard" instructions, or other cont t can cause the final pixel mask to differ from the pixel mask y enabled with kill pixel mode is used by the pixel shader. ha Test Enable or AlphaToCoverage Enable enabled. nerates and outputs oMask. g is fully supported in hardware and therefore not via to need to ENABLE this bit <u>due to ClipDistance clippi</u>	
25	 Pixel Shader Kill Pixel Format: This bit, if ENABLED, indicates (discard) pixels or samples, oth to be ENABLED in the followin The API pixel shader prograthe pixel shader kernel that received on dispatch. A sampler with chroma key Any render target has Alple The pixel shader kernel ge Note: As ClipDistance clipping instructions, there should be not provide the pixel shader computed Depth 	Enable that the PS kernel or color calculator has the ability there than due to depth or stencil testing. This bit is rear g situations: am contains "killpix" or "discard" instructions, or other cont t can cause the final pixel mask to differ from the pixel mask y enabled with kill pixel mode is used by the pixel shader. ha Test Enable or AlphaToCoverage Enable enabled. nerates and outputs oMask. g is fully supported in hardware and therefore not via to need to ENABLE this bit <u>due to ClipDistance clippi</u> Mode numerated Type	


Value	Name	Description			
0h	PSCDEPTH_OFF	Pixel shader does not compute depth			
1h	PSCDEPTH_ON	Pixel shader computes depth with no guarantee as to its value			
2h	PSCDEPTH_ON_GE	Pixel shader computes depth and guarantees that oDepth > = SourceDepth			
3h	PSCDEPTH_ON_LE	Pixel shader computes depth and guarantees that oDepth <= SourceDepth			
		Programming Notes			
When be pro	bit 5 is set in WM_S grammed to values	TATE(i.e. RT independent rasterization is enabled), this field o :: 2h or 3h.			
21 Early D	epth/Stencil Cont	rol			
Forma	t:	U2 Enumerated Type			
This fie	d specifies the beh	avior of early depth/stencil test.			
Value	Name	Description			
0hEDSC_NORMAL1hEDSC_PSEXEC2hEDSC_PREPS		Depth/Stencil Test/Write behaves as if it happens post-shader owever the pixel shader is not necessarily executed if the ixel fails depth or stencil test (this is the legacy behavior)			
		Depth/Stencil Test/Write behaves as if it happens post-shader, and the pixel shader is executed if the pixel fails depth or stencil test (although pre-shader actions such as primitive inclusion, stipple, etc. will still cause the shader not to execute)			
		Depth/Stencil Test/Write behaves as if it happens pre-shader. The pixel shader is not executed if the pixel fails depth or stencil test. Depth and stencil writes occur even if the pixel is killed by the shader or post-shader by alpha test, etc. Depth output by the pixel shader is ignored			
3h	Reserved				
		Programming Notes			
If EDSC	C_PSEXEC mode is s	elected, Thread Dispatch Enable must be set.			
		Restriction			
Restric accura	tion: When value o te.	f "2h" is programmed, PS_INVOCATIONs_COUNT may not be			
D Pixel S	hader Uses Source	Depth			
Forma	t:	Enable			
This bit, if ENABLED, indicates that the PS kernel requires the source depth value (vPos.z) to passed in the payload. The source depth value is interpolated according to the Position ZV Interpolation Mode state					



			3DSTATE_VS					
	Forma	ət:	Enable					
	This bi (vPos.v Interpo	es that the PS kernel requires the interpolated source W value payload. The W value is interpolated according to the Position ZV						
18:17	Positio	on ZW Interpolation	Mode					
	Forma	at:	U2 Enumerated Type					
	This fie coordi detern Requir	eld elects "interpolation nates passed in the P nine whether these co res Depth, Pixel Shade	on mode" associated with the Position Z (source depth) and W S payload when the PS requires Position as input. This field does n pordinates are actually included in the payload (see Pixel Shader er Requires W).					
	Value	Name	Description					
	0h	INTERP_PIXEL	Evaluate Z & W at the pixel center or UL corner (as specified by Pixel Location of 3DSTATE_MULTISAMPLE)					
	1h	Reserved						
	2h	INTERP_CENTROID						
	3h	INTERP_SAMPLE						
	Programming Notes							
	FATE, value of 3h is not defined for this field. bit 5 in dword 1 (RT Independent Rasterization Enable) is set and y) is not set in WM_STATE, value of 3h is not defined for this field.							
16:11	Baryce	arycentric Interpolation Mode						
	Forma	at:	Enable[6]					
Controls v Bit 0: Pers Bit 1: Pers Bit 2: Pers Bit 3: Nor Bit 4: Nor Bit 5: Nor		ols which barycentric i Perspective Pixel Loca Perspective Centroid Perspective Sample b Non-perspective Pixe Non-perspective Cen Non-perspective Sam	interpolation terms must be passed into the pixel shader kernel. ation barycentric is required barycentric is required arycentric is required I Location barycentric is required troid barycentric is required aple barycentric is required					
	Programming Notes							
	If contiguous dispatch modes are enabled, only bit 3 (non-perspective pixel location) can be set, all other bits in this field must be zero. Pixel Location below refers to either the upper left corner or pixel center depending on the Pixel Location state of 3DSTATE_MULTISAMPLING). MSDISPMODE_PERSAMPLE is required in order to select Perspective Sample or Non- perspective Sample barycentric coordinates. Restriction: When Centroid Barycentric mode is required, HW may produce incorrect interpolation results when a 2X2 pixels have unlit pixels.							
10	Pixel S	Shader Uses Input Co	overage Mask					
	Format: Enable This bit, if ENABLED, indicates that the PS kernel requires the input coverage mask to be passed in the payload.							

Command Reference - Instructions



			3D	SIAIE_V	S				
9:8	Line End Cap Antialiasing Region Width								
	Forma	t:					U2		
	This fiel	ld specifies th ted.	ne distances o	ver which the cov	/erage o	of anti-a	liased lir	ne end caps ar	e
	١	/alue	Name		Descrip	otion			
	0h			0.5 pixels					
	1h			1.0 pixels					
	2h			2.0 pixels					
	3h			4.0 pixels					
7:6	Line Ar	ntialiasing Re	egion Width						
	Forma	t:					U2		
	This fie	d specifies th	ne distance ov	er which the anti	-aliased	l line cov	/erage is	computed.	
	١	/alue	Name		Descrip	otion			
	0h			0.5 pixels					
	1h			1.0 pixels					
	2h			2.0 pixels					
	3h			4.0 pixels					
5	5 Reserved Format: MBZ								
4	Polygon Stipple Enable								
	Format:				Enable	9			
	Enables	the Polygon	Stipple functi	on.					
3	Line Stipple Enable								
	Format:				Enable	9			
	Enables	the Line Stip	ple function.						
2	Point R	asterization	Rule						
	Format: 3D_RasterizationRule								
	This fie	d specifies th	ne rasterization	n rules to be app	lied whe	enever t	he edge	s of a point pr	imitiv
	fall exa	ctly on a pixe	l sampling po	int.			-		1
	Value	Na	me		De	scriptio	n		
	0h	RASTRULE_U	JPPER_LEFT	To match "normal" upper left rules for surface primitives					
	1h	RASTRULE_U	JPPER_RIGHT	To match OpenGL point rasterization rules (round to					
				 OpenGL screen 	origin o	of lower	er right (left).	direction wrt	
1:0	Multisa	ample Raster	rization Mode	9					
	Forma	t:	U2 ei	numerated type					
	This fie	d determines	s whether mul	tisample rasteriza	ation is	turned o	on/off, a	nd how the pix	kel
	sample	point(s) are o	defined. Softw	are sets this acco	ording t	o the AF	I, the AF	PI's multisamp	le en



	3DSTATE_VS								
		state setting (if any), and whether 1X or 4X MSRTs are bound. This state is duplicated in 3DSTATE_SF and both must be set to the same value. Refer to the "Multisampling" section for details on the settings of this field.							
		Va	lue	Name					
		0h	MSRASTM	MSRASTMODE_OFF_PIXEL					
		1h	MSRASTM	IODE_OFF	_PATTERN				
		2h	MSRASTM	IODE_ON_	PIXEL				
		3h	MSRASTM	IODE_ON_	PATTERN				
2	31	Multisa	ample Dispatch M	ode					
		Format	t:	U1 Enu	merated Type				
		This bit, along with Number of Multisamples, determines how PS threads are dispate					dispatcheo	d.	
		Softwar	e programs this bi	s this bit depending on the per-pixel v.s per-sample PS execution requirement.					
		When RT Independent Rasterizat			on Enable = 1, value	of Uh for this field is	s not allow	wed.	
		Value	Name			escription			
		Oh	MSDISPMODE_PE	RSAMPLE	This is the high-qualit where (over and abov run for each covered used for "normal" nor 1X), given Number of programmed to NUN	y DX10.1 multisample re PERPIXEL mode) th sample. This mode is n-multisample render Multisamples is ISAMPLES_1.	e mode e PS is also ing (aka		
		1h	1h MSDISPMODE_PERPIXEL		This is the classic multisample mode of operation, typically used for both antialiasing and transparency. Setup and rasterization operate in full multisample mode, testing coverage and depth/stencil test at the sample level but only running the PS once per pixel.				
	30:0	Reserve	ed						
		Format:				MBZ			



add	- Addition

Source:

Length Bias:

The add instruction performs component-wise addition of src0 and src1 and stores the results in dst. Addition of two floating-point numbers follows rules in add (IEEE mode) or add (ALT mode).

Format:

[(pred)] add[.cmod] (exec_size) dst src0 src1

EuIsa

4

Programming Notes

Use a source modifier with add to implement subtraction.

	Syntax	
[(pred)] add[.cmod]	(exec_size) req req req [(pred)] add	[.cmod] (exec_size) reg reg imm32

	Pseudocode						
Evaluate(src0.chan	<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src0.chan[n] + src1.chan[n]; } }</pre>						
Predicatio	n Condition	al Modifier Sat	turation Source Modifier				
Υ	Υ	Y	Υ				
Src Types	c Types Dst Types						
*B,*W,*D	*B,*W,*D						
*B,*W,*D	F						
F	F						
DF	DF						
DWord	Bit		Description				
03	127:64	ImmSource					
		Exists If:	([ImmSource][Src1.RegFile]=='IMM')				
		Format:	ormat: EU_INSTRUCTION_SOURCES_REG_IMM				
	127:64	RegSource					
		Exists If:	([RegSource][Src1.RegFile]!='IMM')				
	Format: EU_INSTRUCTION_SOURCES_REG_REG						
	63:32	Operand Controls					
		Format: EU_INSTRUCTION_OPERAND_CONTROLS					
	31:0	Header					
		Format:	EU_INSTRUCTION_HEADER				



addc - Addition with Carry

Source: EuIsa

Length Bias: 4

The addc instruction performs component-wise addition of src0 and src1 and stores the results in dst; it also stores the carry into acc.

If the operation produces a carry out, 0x00000001 is stored in acc, else 0x00000000 is stored in acc.

Format:

[(pred)] addc[.cmod] (exec_size) dst src0 src1

Restriction

Syntax

Restriction: AccWrEn is required. The accumulator is an implicit destination and thus cannot be an explicit destination operand.

[(pred)] addc[.cmod] (exec_size) reg reg reg [(pred)] addc[.cmod] (exec_size) reg reg imm32

Pse	eud	oco	bde	
F St	euu	ULL	Jue	

Evaluate(WrEn); for $(n = 0)$	<pre>; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n]</pre>	=
<pre>src0.chan[n] + src1.chan[n]</pre>	<pre>; acc.chan[n] = carry(src0.chan[n] + src1.chan[n]); } }</pre>	

	Predication	Conditional	Modifier	Saturation	Source Modifier				
	Υ	Υ		Ν	Ν				
	Src Types D	ost Types							
	UD U	ID							
	DWord	Bit			Description				
	03	127:64	ImmSour	ce					
		Exists If:	If: ([ImmSource][Src1.RegFile]=='IMM')						
	Format		Format:	EU_II	EU_INSTRUCTION_SOURCES_REG_IMM				
127:64 R		RegSourc	RegSource						
			Exists If:	([Reg	Source][Src1.RegFi	e]!='IMM')			
			Format:	at: EU_INSTRUCTION_SOURCES_REG_REG					
63:32 Operand		Operand	and Controls						
Format:		Format:	EU_INSTRUCTION_OPERAND_CONTROLS						
		31:0	Header						
			Format:		EU_INSTRUCTION_H	IEADER			



asr - Arithmetic Shift Right							
Source: EuIsa							
Length Bias: 4							
Description							
Perform component-wise arithmetic right shift of the bits in src0 by the shift count indicated in src1, storing the results in dst. If src0 has a signed type, insert copies of src0's sign bit in the number of MSBs indicated by the shift count. Otherwise insert 0 bits.							
The shift count is taken from the low five bits of src1, regardless of the src1 type and treated as an unsigned integer in the range 0 to 31.							
Format: [(pred)] asr[.cmod] (exec_size) dst src0 src1							
Programming Notes							
If src0 is -1, the result is -1 regardless of the shift count.							
For unsigned src0 types, asr and shr produce the same result.							
Syntax							
[(pred)] asr[.cmod] (exec_size) reg reg reg [(pred)] asr[.cmod] (exec_size) reg reg 1mm32							
Pseudocode							
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.channel[n]) { shiftCnt = src1.chan[n] & 0x1F; // Always use low 5 bits for shift count. if (src0.chan[n] >= 0) { dst.chan[n] = src0.chan[n] >> shiftCnt; } else { int maskLSB = pow(2, shiftCnt) - 1; if (maskLSB & src0.chan[n] == 0) { dst.chan[n] = sign(src0.chan[n]) * ((abs)src0.chan[n] >> shiftCnt); } else { dst.chan[n] = sign(src0.chan[n]) * ((abs)src0.chan[n] >> shiftCnt) - 1; } } }</pre>							
Predication Conditional Modifier Saturation Source Modifier							
Y Y Y Y							
Src TypesDst Types*B,*W,*D*B,*W,*D							
DWord Bit Description							
03 127:64 ImmSource							
Exists If: ([ImmSource][Src1.RegFile]=='IMM')							
Format: EU_INSTRUCTION_SOURCES_REG_IMM							
127:64 RegSource							
Exists If: ([RegSource][Src1.RegFile]!='IMM')							
Format: EU_INSTRUCTION_SOURCES_REG_REG							
63:32 Operand Controls							
Format: EU_INSTRUCTION_OPERAND_CONTROLS							
31:0 Header Format: EU_INSTRUCTION_HEADER							



			avg	g - Average	9		
Source:	EuIsa	1					
Length Bias:	4						
The avg instr integer avera src1 and ther	The avg instruction performs component-wise integer average of src0 and src1 and stores the results in dst. An integer average uses integer upward rounding. It is equivalent to increment one to the addition of src0 and src1 and then apply an arithmetic right shift to this intermediate value.						
Format: The avg instr integer avera src1 and ther	ruction perfo ge uses integ apply an ari	rms compo Jer upward thmetic rig	onent-wise i rounding. I ht shift to tl	nteger average of sr t is equivalent to inc nis intermediate valu	rc0 and src1 and stores the results in dst. An crement one to the addition of src0 and ue.		
				Syntax			
[(pred)] av	rg[.cmod] (exec_size) reg reg	reg [(pred)] avg	g[.cmod] (exec_size) reg reg imm32		
				Pseudocode			
Evaluate(Wr (src0.chan]	En); for ([n] + src1.	n = 0; n chan[n] +	< exec_s: 1) >> 1;	ize; n++) { if // Use arithmet:	(WrEn.chan[n]) { dst.chan[n] = ic shift right. } }		
Predication	Conditional	Modifier	Saturation	Source Modifier			
Υ	Υ		Υ	Υ			
Src Types D	st Types						
*B,*W,*D *E	3,*W,*D						
DWord	Bit			Dese	cription		
03	127:64	ImmSour	ce				
		Exists If:	([Imr	nSource][Src1.RegFi	le]=='IMM')		
		Format:	EU_II	NSTRUCTION_SOUR	CES_REG_IMM		
	127:64	RegSourc	e				
Exists If:		([Reg	gSource][Src1.RegFil	e]!='IMM')			
		Format: EU_INSTRUCTION_SOURCES_REG_REG					
	63:32	Operand	Controls				
		Format:	EU_IN	STRUCTION_OPERA	ND_CONTROLS		
	31:0	Header					
		Format:		EU_INSTRUCTION_F	IEADER		



bfe - Bit Field Extract

Source:			Ει	ıIsa												
Length E	ias: 4															
Compo src1. Sto type). The wid 0x1f. If width If offse than wid extendi	Component-wise extract a bit field from src2 using the bit field width from src0 and the bit field offset from src1. Store the extracted bit field value in the low bits of dst and sign extend (if D type) or zero extend (if UD type). The width and offset values are from the low five bits of src0 and src1 respectively, or src0 & 0x1f and src1 & 0x1f. If width is zero, the result is zero. If offset + width > 32 then the extracted bit field is bits offset to 31 of src2, extracting only 32 - offset bits, less than width as the bit field cannot extend past the MSB of the source value. Otherwise extract width bits extending from bit positions offset to offset + width - 1.															
[(pred)]] bfe ((exed	:_size)	dst src	0 src1	src2										
							Res	strictio	on							
Restrict	ion: N	lo ad	cumu	lator a	ccess, i	mplicit	or ex	plicit.								
Restrict Formate	ion: A s.	All th	ree-sc	urce in	structi	ons hav	ve cer	tain re	strictio	ns, de	scribe	ed in Instructio	on Mac	hine		
								Syr	ntax							
[(pred)] bf	fe (exec_	size)	reg 1	reg reg	g reg	Ī								
Bsaudocoda																
								Pseud	locode							
Evalua src0.c 0x0000 width sign b { if (dst.ch	te(Wi han[r 0000; - off it of src2 an[n]	rEn) n][4 ; } fset f ds 2 is] =	; for :0]; else); if t.cha sign src2.	UD off if ((src2 n } e chan[r	= 0; r fset = (width 2 is s lse { { dst. n] >>	n < exe = srcl. h + off signed) dst.ch .chan[r offset	ec_si chan [set) { d han[n h] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } }</pre>	if f (v st.c] = ds n[n]] >> } }	(WrE width han[r t.cha >> (offs	<pre>Cn.chan[n]) n == 0) { d n] = src2.ch n[n] >> (32 32 - width) set; // pad</pre>	{ UD st.cha an[n] - wic ; // g sign b	width an[n] << (3 lth); pad 0 pit }	= = 2 - // pa } } e else	d lse {
Evalua src0.ci 0x0000 width sign b { if (dst.ch Predica	te(Wi han[r 0000; - off it of src2 an[n]	rEn) n][4 ; } fset f ds 2 is] =	; for :0]; else); if t.cha sign src2.	<pre>(n = UD off if ((src2 n } e ed) chan[1 mal Mo</pre>	= 0; r fset = (width 2 is s lse { { dst. 1] >> difier	<pre>n < exe = src1. n + off signed) dst.ch .chan[r offset</pre>	ec_si chan [set) { d nan[n] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod</pre>	<pre>if f (v st.c] = dst n[n]] >> } }</pre>	(WrH width han[r t.cha >> (offs	<pre>En.chan[n]) h == 0) { d h] = src2.ch nn[n] >> (32 32 - width) set; // pad</pre>	{ UD st.cha an[n] - wic ; // g sign b	width an[n] << (3 dth); bad 0 bit }	= 2 - // pa } } e else	d lse {
Evalua src0.c: 0x0000 width sign b { if (dst.ch Predica Y	te(Wi han[r 0000; - off it of src2 an[n]	rEn) n][4 ; } f set f ds 2 is] = Cor N	; for :0]; else); if t.cha sign src2.	UD off if ((src2 n } el chan[r nal Mo	= 0; r Eset = (width 2 is s Lse { { dst. 1] >> difier	n < exe = srcl. n + off signed) dst.ch .chan[r offset Satura N	ec_si chan [set) { d nan[n 1] = ; //	Pseud ze; n [n][4 < 32	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod</pre>	<pre>if f (v st.c] = ds; n[n]] >> } ifier</pre>	(WrE width han[r t.cha >> (offs	<pre>En.chan[n]) h == 0) { d h] = src2.ch nn[n] >> (32 32 - width) set; // pad</pre>	{ UD st.cha an[n] - wic ; // g sign b	width an[n] << (3 lth); bad 0 bit }	.= = 2 - // pa } } e else	d lse {
Evalua src0.c: 0x0000 width sign b { if (dst.ch Predica Y Src Typ	te(Wi han[r 0000; - off it of src2 an[n]	rEn) n][4 ; } fset f ds 2 is] = Cor N	; for :0]; else); if t.cha sign src2.	UD off if ((src2 n } e: ed) chan[1 nal Mo	= 0; r Eset = (width 2 is s lse { { dst. n] >> difier	n < exe = srcl. n + off signed) dst.ch .chan[r offset Satura N	ec_si chan [set) { d han[n] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad Sourc N	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod</pre>	<pre>if f (v st.cl = ds: n[n]] >> } }</pre>	(WrE width han[r t.cha >> (offs	<pre>En.chan[n]) h == 0) { d h] = src2.ch nn[n] >> (32 32 - width) set; // pad</pre>	{ UD st.cha an[n] - wic ; // g sign b	width an[n] << (3 lth); bad 0 bit }	.= 2 - // pa } } e else	d lse {
Evalua src0.c 0x0000 width sign b { if (dst.ch Predica Y Src Typ UD	te(Wi han[r 0000; - off src2 an[n] htion	rEn) n][4 ; } fset f ds 2 is] = Cor N Dst T	; for :0]; else); if t.cha sigr src2.	<pre>(n = UD off if ((src2 n } e] ed) - chan[n nal Mo</pre>	= 0; r fset = (width 2 is s lse { { dst. 1] >> difier	n < exe = src1. n + off signed) dst.ch .chan[r offset Satura N	ec_si chan [set) { d han[n 1] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod</pre>	<pre>if f (v st.cl = dst n[n]] >> } } ifier</pre>	(WrE width han[r t.cha >> (offs	<pre>En.chan[n]) h == 0) { d h] = src2.ch an[n] >> (32 32 - width) set; // pad</pre>	{ UD st.cha an[n] - wic ; // g sign b	width an[n] << (3 lth); bad 0 bit }	= 2 - // pa } } e else	d lse {
Evalua src0.c 0x0000 width sign b { if (dst.ch Predica Y Src Typ UD D	te(Wi han[r 0000; - off it of src2 an[n] ntion U U D	rEn) n][4 ; } fset f ds 2 is] = Cor N Dst T JD	; for :0]; else); if t.cha sign src2.	<pre>(n = UD off if ((src2 n } e] ed) chan[n nal Mo</pre>	= 0; r fset = (width 2 is s lse { { dst. 1] >> difier	n < exe = srcl. n + off signed) dst.ch .chan[r offset Satura N	ec_si chan [set) { d han[n 1] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod</pre>	<pre>if f (m st.c] = dsf n[n]] >> } }</pre>	(WrE width han[r t.cha >> (offs	<pre>En.chan[n]) n == 0) { d n] = src2.ch an[n] >> (32 32 - width) set; // pad</pre>	{ UD st.cha an[n] - wic ; // <u>p</u> sign b	width an[n] << (3 lth); bad 0 bit }	= 2 - // pa } } e else	d lse {
Evalua src0.ci 0x0000 width sign b { if (dst.ch Predica Y Src Typ UD D DWord	te(Wi han[r 00007 - off it of src2 an[n] ntion U D Bi	rEn) n][4 ; } fset f ds 2 is 2 is 2 is 2 is 2 is 1 = Cor N D St T JD	; for :0]; else); if t.cha sign src2.	<pre>(n = UD off if (0 (src2 n } el chan[r nal Mo</pre>	= 0; r fset = (width 2 is s lse { { dst. 1] >> difier	n < exe = srcl. n + off signed) dst.ch chan[r offset Satura N	ec_si chan [set) { d han[n h] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod</pre>	<pre>if f (m st.cl = ds n[n]] >> } } ifier</pre>	(WrE width han[r t.cha >> (offs	<pre>En.chan[n]) n == 0) { d n] = src2.ch an[n] >> (32 32 - width) set; // pad</pre>	{ UD st.cha an[n] - wic ; // g sign k	width an[n] << (3 lth); bad 0 bit }	. = = 2 - // pa } } e else	d lse {
Evalua src0.ci 0x0000 width sign b { if (dst.ch Predica Y Src Typ UD D D DWord 03	te(Wi han[r 0000; - off it of src2 an[n] tion U D Bi 127::	rEn) n][4 ; } fset f ds 2 is 2 is 2 is 2 is 2 is 1 = Cor N Dst T JD 0 it 126	; for :0]; else); if t.cha sign src2. dition ypes Reser	ved	= 0; r Eset = (width 2 is s lse { { dst. 1] >> difier	n < exe = srcl. n + off signed) dst.ch chan[r offset Satura N	ec_si chan [set) { d han[n h] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod </pre>	<pre>if f (m st.cl = ds n[n]] >> } } ifier</pre>	(WrE width han[r t.cha >> (offs	<pre>En.chan[n]) n == 0) { d n] = src2.ch an[n] >> (32 32 - width) set; // pad </pre>	{ UD st.cha an[n] - wic ; // g sign b	width an[n] << (3 lth); bad 0 bit }	. = = 2 - // pa } } e else	d lse {
Evalua src0.ci 0x0000 width sign b { if (dst.ch Predica Y Src Typ UD D DWord 03	te (Wi han[r 0000; - off it of src2 an[n] tion U D Bi 127::	rEn) n][4; fset f ds 2 is 2 is 2 is 1 = Cor N Dost T JD D 126	; for :0]; else); if t.cha sign src2. ditio	ved	= 0; r Eset = (width 2 is s lse { { dst. n] >> difier	n < exe = srcl. n + off signed) dst.ch chan[r offset Satura N	ec_si chan [set) { d han[n] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod </pre>	<pre>if f (v st.cl = dst n[n]] >> } } ifier</pre>	(WrE width han[r t.cha >> (offs	<pre>En.chan[n]) h == 0) { d h] = src2.ch an[n] >> (32 32 - width) set; // pad MBZ</pre>	{ UD st.cha an[n] - wic ; // g sign k	width an[n] << (3 dth); bad 0 bit }	. = = 2 - // pa } } e else	d lse {
Evalua src0.ci 0x0000 width sign b { if (dst.ch Predica Y Src Typ UD D DWord 03	te (Wi han[r 0000; - off it of src2 an[n] tion U D Bi 127:: 125::	rEn) n][4; fset f ds 2 is 2 is 2 is 2 is 1 = Cor N Dost T JD D D 126 106	; for :0]; else); if t.cha sign src2. dition ypes Reser Form	ved ved ved ved ved ved ved ved	= 0; r Eset = (width 2 is s lse { { dst. 1] >> difier	n < exe = srcl. n + off signed) dst.ch chan[r offset Satura N	ec_si chan [set) { d han[n] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad Sourc N	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod Des RAND_S</pre>	<pre>if f (v st.cl = dst n[n]] >> } ifier script</pre>	(WrE width han[r t.cha >> (offs ion	<pre>En.chan[n]) n == 0) { d n] = src2.ch an[n] >> (32 32 - width) set; // pad MBZ MBZ HREE_SRC</pre>	{ UD st.cha an[n] - wic ; // g sign b	width an[n] << (3 lth); bad 0 bit }	. = = 2 - // pa } } e else	d lse {
Evalua src0.ci 0x0000 width sign b { if (dst.ch Predica Y UD D DWord 03	te (Wi han[r 0000; - off it of src2 an[n] tion tion U D Bi 127:: 125::	rEn) n][4; fset f ds 2 is 2 is 1 = Cor N Dst T JD D it 126 106	; for :0]; else); if t.cha sign src2. dition ypes Reser Form Sourc Form	ved	= 0; r Eset = (width 2 is s lse { { dst. 1] >> difier	n < exe = src1. n + off signed) dst.ch chan[r offset Satura N	cc_si chan (dan[n] = ; //	Pseud ze; n [n][4 < 32 st.ch] = d src2. pad Sourc N	<pre>locode ++) { :0]; i) { d an[n] st.cha chan[n 0 } } ce Mod Des RAND_S</pre>	<pre>if f (v st.cl = dst n[n]] >> } ifier script</pre>	(WrE width han[r t.cha >> (offs ion	<pre>En.chan[n]) h == 0) { d h] = src2.ch an[n] >> (32 32 - width) set; // pad MBZ MBZ HREE_SRC</pre>	{ UD st.cha an[n] - wic ; // g sign k	width an[n] << (3 dth); bad 0 bit }	. = = 2 - // pa } } e else	d lse {



		bf	fe - Bit Field Extract				
104:85	Source 1						
	Format:	EU_INS	TRUCTION_OPERAND_SRC_REG_THREE_SRC				
84	Reserved						
	Format:		MBZ				
83:64	Source 0						
	Format:	EU_INS	TRUCTION_OPERAND_SRC_REG_THREE_SRC				
63:56	Destination	Register	Number				
	Format:		DstRegNum				
55:53	Destination	Subregis	ster Number				
	Format:		DstSubRegNum[2:0]				
52:49	Destination	Channel	Enable				
	Format:		ChanEn[4]				
	Four channel enables are defined for controlling which channels are written into the destination region. These channel mask bits are applied in a modulo-four manner to all ExecSize channels. There is 1-bit Channel Enable for each channel within the group of 4. If the bit is cleared, the write for the corresponding channel is disabled. If the bit is set, the write is enabled. Mnemonics for the bit being set for the group of 4 are <i>x</i> , <i>y</i> , <i>z</i> , and <i>w</i> , respectively, where <i>x</i> corresponds to Channel 0 in the group and <i>w</i> corresponds to channel 3 in the group						
48	Reserved						
	Format:		MBZ				
47	NibCtrl						
	Format:		NibCtrl				
46	Reserved						
	Format:		MBZ				
45:44	Destination Data Type This field contains the data type for the destination						
	Valu	е	Name				
	00b		Single Precision Float				
	01b		DWord				
	10b		Unsigned DWord				
	11b Double Precision Float						
43:42	Source Data Type This field contains the data type for all three sources						
	Valu	е	Name				
	00b		Single Precision Float				
	01b		DWord				
	10b		Unsigned DWord				
	11b		Double Precision Float				
41:40	Source 2 Mc	difier					



		bf	e - Bit Field Extract				
	Exists If: ([Property[Source Modification]=='true')						
	Format:	SrcM	lod				
39:38	Source 1 Mod	Source 1 Modifier					
	Exists If:	([Pro	perty[Source Modification]=='true')				
	Format:	SrcN	lod				
41:36	Reserved						
	Exists If:	([Pro	perty[Source Modification]=='false')				
	Format:	MBZ					
37:36	Source 0 Modifier						
	Exists If:	xists If: ([Property[Source Modification]=='true')					
	Format:	SrcM	lod				
35	Reserved						
	Format:		N	1BZ			
34	Flag Register Number This field contains the flag register number for instructions with a non-zero Conditional Modifier.						
33	Flag Subregister Number This field contains the flag subregister number for instructions with a non-zero Conditional Modifier.						
32	Reserved						
	Format:		Ν	1BZ			
31:0	Header						
	Format: EU_INSTRUCTION_HEADER						



bfi1 - Bit Field Insert 1 Source: EuIsa 4 Length Bias: The bfi1 instruction is the first instruction in a two-instruction macro for bfi (Bit Field Insert). The bfi1 instruction component-wise generates mask with control from src0 and src1 and stores the results in dst. The mask is used in the bfi2 instruction to generate the final result of bfi. Create a bit mask corresponding to the bit field width and offset in src0 and src1. Store the bit mask in dst. The mask has all bits in the bit field set to 1 and all other bits as 0. The width and offset values are from the low five bits of src0 and src1 respectively, or src0 & 0x1f and src1 & 0x1f. If width is zero, the result is zero. The bfi macro has four source operands: src0 - bit field width in low five bits, src1 - bit field offset/starting bit position in low five bits, src2 - bit field value to insert, using only the number of least significant bits given by width in src0, and src3 - overall value into which the bit field is inserted, providing all bits other than the inserted bits for the result value. bfi dst src0 src1 src2 src3 // Translates to these two instructions: bfi1 dst src0 src1 bfi2 dst dst src2 src3 Format: [(pred)] bfi1 (exec_size) dst src0 src1 **Programming Notes** No accumulator access, implicit or explicit. **Syntax** [(pred)] bfil (exec_size) reg reg reg [(pred)] bfil (exec_size) reg reg imm32 **Pseudocode** Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { UD width =</pre> src0.chan[n][4:0]; UD offset = src1.chan[n][4:0]; dst = ((1 << width) - 1) << offset; } }</pre> Predication Conditional Modifier Saturation Source Modifier Υ Ν Ν Ν Src Types **Dst Types** UD UD D D DWord Bit Description



bfi1 - Bit Field Insert 1								
03	127:64	ImmSource						
		Exists If:	([ImmSource][Src1.RegFile]=='IMM')					
		Format:	EU_INSTRUCTION_SOURCES_REG_IMM					
	127:64	RegSource	urce					
		Exists If:	([RegSource][Src1.RegFile]!='IMM')					
		Format:	EU_INSTRUCTION_SOURCES_REG_REG					
	63:32	Operand Cont	rols					
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS					
	31:0	Header	·					
		Format:		EU_INSTRUCTION_HEADER				



bfi2 - Bit Field Insert 2

Source: EuIsa 4 Length Bias: The bfi2 instruction is the second instruction in a two-instruction macro for bfi (Bit Field Insert). The bfi2 instruction component-wise performs the bitfield insert operation on src1 and src2 based on the mask in src0. Use the mask in src0 to take a bit field value from the low bits of src1 and combine it with the value from src2 (so src2 provides all bits other than those masked out and replaced by the bit field value). Store the result in dst. The bfi macro has four source operands: src0 - bit field width in low five bits, src1 - bit field offset/starting bit position in low five bits, src2 - bit field value to insert, using only the number of least significant bits given by width in src0, and src3 - overall value into which the bit field is inserted, providing all bits other than the inserted bits for the result value. bfi dst src0 src1 src2 src3 // Translates to these two instructions: bfi1 dst src0 src1 bfi2 dst dst src2 src3 Format: [(pred)] bfi2 (exec size) dst src0 src1 src2 Restriction Restriction: No accumulator access, implicit or explicit. Restriction: All three-source instructions have certain restrictions, described in Instruction Machine Formats. **Syntax** [(pred)] bfi2 (exec_size) reg reg reg reg **Pseudocode** Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { UD offset =</pre> LZD(reverse(src0.chan[n]))-1; // offset is the number of LSB zero bits below the bit mask which has all 1s. // width (implied by the logic) is the number of 1 bits in the mask value, which should be all 1s. dst.chan[n] = ((src1.chan[n] << offset) & src0.chan[n]) | (src2.chan[n] & ! src0.chan[n]); } **Predication Conditional Modifier Saturation Source Modifier** Ν Ν Ν Src Types Dst Types

Description

Y

UD

DWord

D

UD D

Bit



		b	fi2 - Bit Field Insert 2					
03	127:126	Reserved						
		Format:	MBZ					
	125:106	Source 2						
		Format: EU_IN	ISTRUCTION_OPERAND_SRC_REG_THREE_SRC					
	105	Reserved						
		Format:	MBZ					
	104:85	Source 1						
		Format: EU_IN	ISTRUCTION_OPERAND_SRC_REG_THREE_SRC					
	84	Reserved						
		Format:	MBZ					
	83:64	Source 0						
		Format: EU_IN	ISTRUCTION_OPERAND_SRC_REG_THREE_SRC					
	63:56	Destination Regist	er Number					
		Format:	DstRegNum					
	55:53	Destination Subre	jister Number					
		Format: DstSubRegNum[2:0]						
	52:49	Destination Chann	el Enable					
		Format:	ChanEn[4]					
		Four channel enables are defined for controlling which channels are written into the						
		ExecSize channels. There is 1-bit Channel Enable for each channel within the group of 4. If the						
		bit is cleared, the write for the corresponding channel is disabled. If the bit is set, the write is						
		enabled. Mnemonics for the bit being set for the group of 4 are x, y, z, and w, respectively,						
		where x corresponds to Channel 0 in the group and w corresponds to channel 3 in the group						
	48	Reserved						
		Format:	MBZ					
	47	NibCtrl						
		Format: NibCtrl						
	46	Reserved						
		Format:	MBZ					
	45:44	Destination Data T	уре					
		This field contains t	ne data type for the destination					
		Value	Name					
		00b	Single Precision Float					
		01b	DWord					
		10b	Unsigned DWord					
		11b	Double Precision Float					
	43:42	Source Data Type						



			bfi	2 - Bit Field Insert 2					
		This field cont	contains the data type for all three sources						
		Value		Name					
		00b		Single Precision Float					
		01b		DWord					
		10b		Unsigned DWord					
		11b		Double Precision Float					
4	41:40	Source 2 Mod	lifier						
		Exists If:	([Prc	operty[Source Modification]=='true')					
		Format:	SrcN	Лоd					
3	39:38	Source 1 Mod	difier						
		Exists If: ([Property[Source Modification]=='true')							
		Format:	Format: SrcMod						
4	41:36	Reserved							
		Exists If:	([Pro	([Property[Source Modification]=='false')					
		Format: MBZ							
3	37:36	Source 0 Mod	difier						
		Exists If: ([Property[Source Modification]=='true')							
		Format:	SrcN	rcMod					
	35	Reserved							
		Format: MBZ							
	34	Flag Register Number This field contains the flag register number for instructions with a non-zero Conditional Modifier.							
	33	Flag Subregister Number This field contains the flag subregister number for instructions with a non-zero Conditional Modifier.							
	32	Reserved							
		Format:		MBZ					
	31:0	Header							
		Format:		EU_INSTRUCTION_HEADER					



bfrev - Bit Field Reverse

Source:

Length Bias:

The bfrev instruction component-wise reverses all the bits in src0 and stores the results in dst.

Format:

[(pred)] bfrev (exec_size) dst src0

Restriction

Syntax

Restriction: No accumulator access, implicit or explicit.

EuIsa 4

[(pred)] bfrev (exec_size) reg reg [(pred)] bfrev (exec_size) reg imm32

Pseudocode	
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { fo idx < 32; idx++) { dst.chan[n][idx] = src0.chan[n][31-idx]; } }</pre>	r (idx = 0;

Predication	Conditional Modifier	Saturation	Source Modifier
Y	Ν	N	N

Src Types Dst Types

UD	UD							
DWord	Bit		Description					
03	127:64	ImmSource						
		Exists If:	([Operand Controls][Src0.RegFile]=='IMM')					
		Format:	EU_INSTRUCTION_SOURCES_IMM32					
	127:64	RegSource						
		Exists If:	([Operand Controls][Src0.RegFile]!='IMM')					
		Format:	EU_INSTRUCTION_SOURCES_REG					
	63:32 Operand Controls		ntrols					
Format:		Format:	at: EU_INSTRUCTION_OPERAND_CONTROLS					
	31:0	Header						
		Format:	EU_INSTRUCTION_HEADER					



		br	c - Bra	nch Conv	erging			
Source:	Eu	uIsa						
Length Bias	Length Bias: 4							
Description								
The brc instruction redirects the execution forward or backward to the instruction pointed by (current IP + offset). The jump will occur if all channels are branched away. UIP should reference the instruction where all channels are expected to come together. JIP should reference the end of the innermost conditional block.								
In GEN bin where reg3 (for examp	In GEN binary, JIP and UIP are at location src1 when immediates and at location src0 when reg32, where reg32 is accessed as a scalar DWord containing both JIP and UIP. The null register must be used (for example, by the assembler) as dst. When offsets are immediate, src0 must be null.							
Format: [(pred)] br	c (exec_size)	JIP UIP						
			Rest	riction				
Restriction:	A brc instru	iction must us	e the Switch	n instruction option				
			S	vntax				
[(pred)]	brc (exec_	_size) imm16	imm16 [(p	pred)] brc (exec	_size) reg32	2		
				Decude code				
Fvaluate(WrEn): for	r(n = 0; r	< 32: n+-	+) { if (WrEn[nl) { patp	nl – TD + UTD: }	olco (
PcIP[n] =	IP + 1; }	<pre></pre>	PcIP != 1	IP + 1) { // fo	or all channe	els Jump(IP + JIP)	; }	
Predicatio	n Conditio	nal Modifier	Saturation	Source Modifier	Source Type	S		
Υ	Ν		Ν	Ν	D			
DWord	Bit			Desc	ription			
03	127:112	UIP						
		Format:				S15		
		The jump dis	stance in nui	mber of eight-byte	units if a jump	is taken for the chan	nel.	
	111:96	JIP						
		Format:				S15		
		The jump dis	stance in nui	mber of eight-byte	units if a jump	is taken for the instru	uction.	
	95:64	Reserved						
		Format:			MBZ	7		
	63:32	Operand Co	ntrol					
		Format:	EU_INST	RUCTION_OPERAN				
	31:0	Header						
		Format:	EU	_INSTRUCTION_HE	ADER			



		b	rd - Br	anch Dive	rgin	g		
Source:		EuIsa						
Length E	ength Bias: 4							
	Description							
The brd	instruct	ion redirects the exe	cution forwa	rd or backward to t	the instru	uction pointed by (current		
IP + off	set). The	jump will occur if ar	iy channels a	re branched away.			-	
In GEN accesse	binary, J d as a so	IP is at location src1 calar DWord. The nul	when immed I register mu	liate and at location st be used at dst lo	n src0 wł cations.	hen reg32, where reg32 is	-	
Format: [(pred)]] brd (ex	ec_size) JIP						
			Restr	iction			_	
Restrict	ion: A bı	d instruction must u	se the Switch	instruction option	•			
-			6.					
[(prod) l brd	(ourog gigo) imm1(Sy	ntax				
[(pred		(exec_size) innuit	b [(pred)]	bid (exec_size)	regsz			
				Pseudocode				
Evalua PcIP[n	te(WrEr] = IP	<pre>1); for (n = 0; n + 1; } } if (any</pre>	n < 32; n++ 7 PcIP == E) { if (WrEn[xIP + JIP) { /	n]) { / any c	PcIP[n] = IP + JIP; } hannel Jump(ExIP + JIP;	else {); }	
Predica	tion Co	onditional Modifier	Saturation	Source Modifier				
Y	Ν		N	Ν				
Src Typ	es							
D								
DWord	Bit			Descript	ion			
03	127:112	Reserved						
		Format:			Ν	MBZ		
	111:96	JIP					1	
		Format:				S15		
		Jump Target Offset. The relative offset in 64-bit units if a jump is taken for the instruction.						
	95:91	Reserved					1	
		Format:	Format: MBZ					
	90	Flag Register Nur Added a second fla	n ber ag register					
	89	Flag Subregister I This field specifies registers in the flag The selected flag instruction. It is the for the instruction.	Added a second flag register Flag Subregister Number This field specifies the sub-register number for a flag register operand. There are two sub- registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and					



brd - Branch Diverging								
	conditional destination, if both predication and conditional modifier are enabled.							
88:64	Source 0							
	Exists If:	(Structu	ure[EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align1')					
	Format:	EU_INST	TRUCTION_OPERAND_SRC_REG_ALIGN1					
88:64	Source 0	-						
	Exists If:	(Structu	[EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align16')					
	Format:	EU_INST	TRUCTION_OPERAND_SRC_REG_ALIGN16					
63:32	Operand (Control						
	Format: EL		EU_INSTRUCTION_OPERAND_CONTROLS					
31:0	Header							
	Format:		EU_INSTRUCTION_HEADER					





			bre	ak - Break	C					
Source:	Eu	uIsa								
Length Bias:	4									
			Des	cription						
The break in switch block	The break instruction is used to early-out from the inner most loop, or early out from the inner most switch block.									
When used channels er referenced should be t	When used in a loop, upon execution, the break instruction terminates the loop for all execution channels enabled. If all the enabled channels hit the break instruction, jump to the instruction referenced by JIP. JIP should be the offset to the end of the inner most conditional or loop block, UIP should be the instruction of the loop block.									
If SPF is ON	N, the UIP m	ust be used to	o update IP;	JIP is not used in th	nis case					
The followin signed 16-b must be of	The following table describes the two 16-bit instruction pointer offsets. Both the JIP and UIP are signed 16-bit numbers, added to IP pre-increment. In GEN binary, JIP and UIP are at location src1 and must be of type W (signed word integer).									
Format: [(pred)] bre	eak (exec_siz	ze) JIP UIP								
			Synta	x						
[(pred)]]	break (exe	ec_size) imm	16 imm16							
				Pseudocode						
Evaluate() + UIP; el; JIP); }	WrEn); for se { PcIP[n] = IP + 1	<pre>exec_si ; } } if (</pre>	ze; n++) { if PcIP != (IP +)	(WrEn.ch 1)) { //	annel[n]) { Po all channels 5	cIP[n] = I Jump(IP +	IP		
Predication	n Conditio	nal Modifier	Saturation	Source Modifier						
Υ	Ν		Ν	Ν						
DWord	Bit			Descr	iption					
03	127:112	UIP								
		Format: S15								
		The jump dis	stance in nur	nber of eight-byte	units if a ju	ımp is taken for th	e channel.			
-	111.06									
	111.90	Format				\$15				
		The jump dis	stance in nur	nber of eight-byte	units if a ju	imp is taken for th	e instructio	n.		
-	95:64	Reserved								
		Format:			١	MBZ				
-	63:32	Operand Co	ntrol							
		Format:	EU_INST	RUCTION_OPERAN	D_CONTRO	DLS				



break - Break				
	31:0	Header		
		Format:	EU_INSTRUCTION_HEADER	



call - Call						
Source:	EuIsa					
Length Bias:	4					
	Description					
The call instruction jumps to a subroutine. It can be predicated or non-predicated. If non- predicated, all enabled channels jump to the subroutine. If predicated, only the channels enabled by PMask jump to the subroutine; the rest of the channels move to the next instruction after the call instruction. If none of the channels jump into the subroutine, the call instruction is treated as a nop. In case of a jump, the call instruction stores the return IP onto the first DWord of the destination register and stores the CallMask in the second DWord of the destination register.						
When SPF is on, the predication control must be scalar.						
The following table describes JIP, the jump offset. JIP must be a signed immediate operand. When a jump occurs, this value is added to IP pre-increment.						
Format: [(pred)] call (exec_siz	ze) dst JIP					

Restriction					
Restriction: The call instruction must have DWord source and destination type, and the destination must be QWord aligned.					
Restriction: The source0 regioning control must be < 2;2,1 > .					
Restriction: The execution size must be 2.					

Sy	ntax
[(pred)] call (exec_size) reg imm16	

Pseudocode
Evaluate(WrEn); for ($n = 0$; $n < exec_size$; $n++$) { if (WrEn.chan[n]) { PcIP[n] = IP + JIP; CallMask[n] = 1; }
else { PcIP[n] = IP + 1; CallMask[n] = 0; } if (PcIP[n] != (IP + 1)) { // any channel jumped dst.chan[0] = IP
+ 1; dst.chan[1] = CallMask; Jump(IP + JIP); }

Predication	Conditional Modifier	Saturation	Source Modifier
Υ	Ν	Ν	Ν

		I	Ost Types			
D,UD						

DWord	Bit	Description
03	127:112	Reserved



DWord	Bit	Description						
		Format:			N	1BZ		
	111:96	JIP						
		Format:				S15		
Jump Target Offset. The relative offset in 64-bit units if a jump is taken for t instruction.						f a jump is taken for the		
	95:91	Reserved						
		Format:			N	1BZ		
	90	0 Flag Register Number Added a second flag register						
	89 Flag Subregister Number This field specifies the sub-register number for a flag register operand. There sub-registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is e the instruction. It is the destination to store conditional flag bits if conditional is enabled for the instruction. The same flag sub-register can be both the pri- source and conditional destination, if both predication and conditional mod							
	88:64	Source 0				1		
		Exists If:	(Struc	cture	[EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align1')		
		Format:	Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1					
	88:64	Source 0						
		Exists If:	: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align16')					
		Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16						
	63:32	Operand	Contr	ol		1		
		Format:		EU_I	NSTRUCTION_OPERAND_CONTRO	DLS		
	31:0	Header						
		Format:			EU_INSTRUCTION_HEADER			



COLOR_BLT

Source: BlitterCS

2

Length Bias:

COLOR_BLT is the simplest BLT operation. It performs a color fill to the destination (with a possible ROP). The only operand is the destination operand which is written dependent on the raster operation. The solid pattern color is stored in the pattern background register.

This instruction is optimized to run at the maximum memory write bandwidth.

The typical Raster operation code = F0 which performs a copy of the pattern background register to the destination.

DWord	Bit	Description						
0	31:29	Client						
		Default Value:	02h 2D Proc	02h 2D Processor				
BR00		Format:		Opcode				
	28:22	Instruction Target	Instruction Target(Opcode)					
		Default Value:			40h			
		Format:			Opcode			
	21:20	32bpp Byte Mask This field is only use	d for 32bpp.					
		Value		N	ame			
		1xb Write Alpha Channel						
		x1b Write RGB Channel						
	19:6	Reserved						
		Format:			MBZ			
	5:0	DWord Length						
		Default Value:				03h		
1	31:26	Reserved						
8840		Format:			MBZ			
RK13	25:24	Color Depth						
		Value		Name				
		00b	8 Bit Color					
		01b	16 Bit Color(565)					
		10b	16 Bit Color(1555)					
		11b	32 Bit Color					
	23:16	Raster Operation						
	15:0	Destination Pitch (Signed) Destination pitch in bytes (Same as before).						



			COLOR_BLT			
2	31:16	Destination H	leight (in scan lines)			
BR14	15:0	Destination E	Destination Byte Width (in bytes)			
3	31:0	Destination Address				
		Format:	GraphicsAddress[31:0]			
BR09		Address of the	e first byte to be written.			
4	31:0	Solid Pattern	Solid Pattern Color			
BR16		0 DIL – [7.0], 1	0 bit – [13.0], 32 bit – [31.0]			



cmp - Compare

Source: EuIsa

Length Bias: 4 The cmp instruction performs component-wise comparison of src0 and src1 and stores the results in the selected flag register and in dst. It takes component-wise subtraction of src0 and src1, evaluating the conditional code (excluding NS signal) based on the conditional modifier, and storing the conditional bits in bit-packed form in the destination flag register and all bits of dst channels. If the dst is not null, for the enabled channels, then all bits of the destination channel will contain the flag value for the channel. When the instruction operates on packed word format, one general register may store up to 16 such comparison results. In DWord format, one general register may store up to 8 results.

A conditional modifier must be specified; the conditional modifier field cannot be 0000b. The comparison does not use the NS (NaN source) signals, as described in the Creating Conditional Flags section. Accordingly the conditional modifier should not be .u (unordered).

For each enabled channel 0b or 1b is assigned to the appropriate flag bit and 0/all zeros or all ones (e.g, byte 0xFF, word 0xFFFF, DWord 0xFFFFFFF) is assigned to dst.

When any source type is floating-point, the cmp instruction obeys the rules described in the tables in the Floating Point Modes section of the Data Types chapter.

Format:

[(pred)] cmp[.cmod] (exec_size) dst src0 src1

Restriction

Restriction: Accumulator cannot be destination, implicit or explicit. The destination must be a general register or the null register.

Restriction: A SIMD16 instruction is not allowed for DWord data types. Use two SIMD8 instructions.

Restriction: If the destination is the null register, the {Switch} instruction option must be used.

S١	'n	ta	X

[(pred)] cmp[.cmod] (exec_size) reg reg reg [(pred)] cmp[.cmod] (exec_size) reg reg imm32

Pseudocode

Evaluate(WrEn); for (n = 0; n < exec_size; n++) { bitMask[n] = 0; if (WrEn.chan[n]) {
results[n] = src0.chan[n] - src1.chan[n]; bitMask[n] = Condition(results[n]); dst.chan[n]
= bitMask[n]; // All bits for dst channel } } flag# = bitMask;</pre>

Predicatio	n Conditio	nal Modifier	Saturation	Source Modifier
Y	Y		N	Υ
Src Types	Dst Types			
*B,*W,*D	*B,*W,*D			
*B,*W,*D	F			
F	F			
DF	DF			



DWord	Bit	Description				
03	127:64	ImmSource				
		Exists If:	([ImmSource][Src1.RegFile]=='IMM')			
		Format:	EU_INSTRUCTION_SOURCES_REG_IMM			
	127:64	RegSource	egSource			
		Exists If:	([RegSource][Src1.RegFile]!='IMM')			
		Format:	EU_INSTRUCTION_SOURCES_REG_REG			
	63:32	Operand Contr	perand Controls			
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS			
	31:0	Header				
		Format:	EU_INSTRUCTION_HEADER			



cmpn - Compare NaN

				Compare		
Source:	E	uIsa				
Length Bias	: 4					
The cmpn results in the the conditi bit-packed channels, t instruction In DWord f	instruction p he selected onal signals form in the hen all bits operates or format, one	performs comp flag register an including NS destination fla of the destinat n packed word general regist	ponent-wise nd in dst. It t based on the ag register a ion channel format, one er may store	special-NaN comp akes component-w e conditional modif nd all bits of dst ch will contain the flag general register m up to 8 results.	arison of src0 and src1 and stores th rise subtraction of src0 and src1, eval fier, and storing the conditional flag annels. If the dst is not null, for the e g value for the channel. When the nay store up to 16 such comparison r	e luating bits in enabled results.
A conditio about the o	nal modifier conditional s	r must be spec signals used is	ified; the co in the Creat	nditional modifier f ing Conditional Fla	ïeld cannot be 0000b. More informa gs section.	tion
For each e 0xFF, word	enabled char 0xFFFF, DW	nnel 0b or 1b i /ord 0xFFFFFf	s assigned to F) is assigne	o the appropriate fl d to dst.	ag bit and 0/all zeros or all ones (e.g	ı, byte
Min/Max i Point Num	nstructions bers section	use cmpn to s 1 for details).	elect the des	stination from the in	nput sources (see the Min Max of Flo	oating
Format: [(pred)] cn	npn[.cmod]	(exec_size) dst	src0 src1			
			Res	striction		
Restriction register or	: Accumulate the null reg	or cannot be c ister.	lestination, i	mplicit or explicit. T	he destination must be a general	
Restriction	: A SIMD16	instruction is r	not allowed f	or DWord data typ	es. Use two SIMD8 instructions.	
Restriction	: If the desti	nation is the n	ull register, t	he {Switch} instruc	tion option must be used.	
				Syntax		
[(pred)] imm32	cmpn[.cmod	d] (exec_siz	e) reg reg	reg [(pred)] c	<pre>mpn[.cmod] (exec_size) reg reg</pre>	3
				Pseudocode		
Evaluate(results[r dst.chan[WrEn); for n] = src0.c [n][0] = b:	r (n = 0; n chan[n] - sr itMask[n]; /	<pre>c < exec_si c1.chan[n] / All bits</pre>	<pre>ze; n++) { bit ; bitMask[n] = for dst channe</pre>	<pre>Mask[n] = 0; 1f (WrEn.chan[n. ConditionNaN(results[n]); 1 } flag# = bitMask;</pre>) {
Predicatio	on Conditio	nal Modifier	Saturation	Source Modifier		
Υ	Y		Ν	Υ		
Src Types	Dst Types					
*B,*W,*D	*B,*W,*D					
*B,*W,*D	F					
F	F					
DF	DF					
DWord	Bit			Des	cription	

Г



		cmp	on -	· Compare NaN
03	127:64	ImmSource		
		Exists If:	([Im	mSource][Src1.RegFile]=='IMM')
		Format:	EU_I	NSTRUCTION_SOURCES_REG_IMM
	127:64	RegSource		
		Exists If:	([Re	gSource][Src1.RegFile]!='IMM')
		Format:	EU_I	INSTRUCTION_SOURCES_REG_REG
	63:32	Operand Controls		
		Format:	EU_IN	ISTRUCTION_OPERAND_CONTROLS
	31:0	Header		
		Format:		EU_INSTRUCTION_HEADER



sendc - Conditional Send Message

Source:

Length Bias:

The sendc instruction has the same behavior as the send instruction except the following.

sendc first checks the dependent threads inside the Thread Dependency Register. There are up to 8 dependent threads in the TDR register. The sendc instruction executes only when all the dependent threads in the TDR register are retired.

Wait for dependencies in the TDR Register to clear, then send a message stored in registers starting at src to a shared function identified by exdesc along with control from desc with a general register writeback location at dst.

Format:

[(pred)] sendc (exec_size) dst src0 exdesc desc

EuIsa

4

Restriction

Restriction: The sendc instruction has the same restrictions as the send instruction.

Pseudocode

if (TDR[7] ... || TDR[2] || TDR[1] || TDR[0]) { wait; } Evaluate(WrEn); MsgChEnable =
WrEn; SourceReg = src0.RegNum; MessageEnqueue(MsgChEnable, ResponseReg, SourceReg, desc,
exdesc);

Predication	Conditional Modifier	Saturation	Source Modifier
Υ	N	Ν	Ν

DWord	Bit	Description						
03	127:96	Message						
		Format:	EU_INS	STRUCTION_OPERAND_SEND_MSG				
	95:89	Flags						
		Format:		EU_INSTRUCTION_FLAGS				
	88:64	Source 0						
		Exists If:	(Structure[E	EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align1')				
		Format:	ormat: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1					
	88:64	Source 0	Source 0					
		Exists If:	Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align16')					
		Format:	ormat: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16					
	63:32	Operand Co	perand Control					
		Format:	ormat: EU_INSTRUCTION_OPERAND_CONTROLS					
	31:28	Controls B	Controls B					
		Format:	Format: EU_INSTRUCTION_CONTROLS_B					
	27:24	Shared Fun	Shared Function ID (SFID) Format:					
		Format:						
	23:8	Controls A						
		Format:	EU_	_INSTRUCTION_CONTROLS_A				



	sendc - Condition	nal Send M	lessage
7	Reserved		
	Format:		MBZ
6:0	Opcode		
	Format:	EU_OPCODE	



cont - Continue Source: EuIsa Length Bias: 4 **Description** The cont instruction disables execution for the subset of channels for the remainder of the current loop iteration. Channels remain disabled until right before the while instuction or right before the condition check code block for the while instruction. If all enabled channels hit this instruction, jump to the instruction referenced by JIP where execution continues. UIP should always reference the loop's associated while instruction. JIP should point to the last instruction of the inner most conditional block if the cont instruction is inside a conditional block. In case of the break instruction directly under the loop, the JIP and the UIP are the same. If SPF is ON, the UIP must be used to update IP; JIP is not used in this case. The following table describes the two 16-bit instruction pointer offsets. Both the JIP and UIP are signed 16-bit numbers, added to IP pre-increment. In GEN binary, JIP and UIP are at location src1 and must be of type W (signed word integer). Format: [(pred)] cont (exec size) JIP UIP Restriction Restriction: The execution size must be the same for the while, break, and cont instructions of the same code block. **Syntax** [(pred)] cont (exec_size) imm16 imm16 **Pseudocode** Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.channel[n]) { if (PMask[n]</pre>) { // PMask is for all channels enabled for the cont instruction. PcIP[n] = IP + UIP; } else { PcIP[n] = IP + 1; } } } for (n = exec_size; n < 32; n++) { PcIP[n] = IP + 1; } if (PcIP != (IP + 1)) { // all channels true Jump(IP + JIP); } Predication Conditional Modifier Saturation Source Modifier γ Ν Ν Ν DWord **Description** Bit 0..3 127:112 UIP Format: S15 The jump distance in number of eight-byte units if a jump is taken for the channel. 111:96 JIP Format: S15

The jump distance in number of eight-byte units if a jump is taken for the instruction.



		СО	nt - Continue	
95:64	Reserved			
	Format:			MBZ
63:32	Operand Cont	rol		
	Format:	EU_I	STRUCTION_OPERAND_CONTR	ROLS
31:0	Header			
	Format:		EU_INSTRUCTION_HEADER	



cbit - Count Bits Set

Source:

Length Bias:

The cbit instruction counts component-wise the total bits set in src0 and stores the resulting counts in dst. Format:

[(pred)] cbit (exec_size) dst src0

Restriction

Restriction: No accumulator access, implicit or explicit.

EuIsa 4

Syntax

[(pred)] cbit (exec_size) reg reg [(pred)] cbit (exec_size) reg imm32

Pseudocode

Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { UD cnt = 0; UD
val = src0.chan[n]; while (val) { if (val & 1) { cnt ++; } val = val >> 1; }
dst.chan[n] = cnt; }

Predication	Conditional Modifier	Saturation	Source Modifier
Y	Ν	Ν	Ν

Src Types Dst Types

00/011/00					
DWord	Bit	Description			
03	127:64	ImmSource	ImmSource		
		Exists If:	([Oper	and Controls][Src0.RegFile]=='IMM')	
		Format:	EU_INS	STRUCTION_SOURCES_IMM32	
	127:64	RegSource			
		Exists If:	([Operand Controls][Src0.RegFile]!='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_REG		
	63:32	Operand Cont	Operand Controls		
		Format:	EU_IN	STRUCTION_OPERAND_CONTROLS	
	31:0	Header			
		Format:	E	EU_INSTRUCTION_HEADER	



dp2 - Dot Product	2
-------------------	---

Source: EuIsa

Length Bias: 4

The dp2 instruction performs a two-wide dot product on four-tuple vector basis and storing the same scalar result per four tuple to all four channels in dst. This instruction is similar to dp4 except that every third and fourth element of src0 (post-source-swizzle if present) are not involved in the computation.

The dot product of two vectors of equal length is the sum of the products of each pair of corresponding elements.

The dp4 instruction includes all four elements of each vector in the dot product. The dp3 instruction includes the first three elements of each vector in the dot product.

Format:

[(pred)] dp2[.cmod] (exec_size) dst src0 src1

Restriction

Restriction: Execution size cannot be less than 4.

Restriction: Horizontal strides must be 1.

Restriction: Source operands cannot be accumulators.

Syntax

[(pred)] dp2[.cmod] (exec_size) reg reg reg [(pred)] dp2[.cmod] (exec_size) reg reg imm32

Pseudocode

Evaluate(WrEn); for (n = 0; n < exec_size; n += 4) { fTmp = src0.chan[n] * src1.chan[n]</pre> + src0.chan[n+1] * src1.chan[n+1]; if (WrEn.chan[n]) dst.chan[n] = fTmp; if (WrEn.chan[n+1]) dst.chan[n+1] = fTmp; if (WrEn.chan[n+2]) dst.chan[n+2] = fTmp; if (WrEn.chan[n+3]) dst.chan[n+3] = fTmp; }

Predication	Conditional Modifier	Saturation	Source Modifier
γ	Y	Υ	Υ

Src Types Dst Types F

Bit		Description	
127:64	ImmSource		
	Exists If:	([ImmSource][Src1.RegFile]=='IMM')	
	Format:	EU_INSTRUCTION_SOURCES_REG_IMM	
127:64	RegSource		
	Exists If:	([RegSource][Src1.RegFile]!='IMM')	
	Format:	EU_INSTRUCTION_SOURCES_REG_REG	
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
	Bit 127:64 127:64 63:32	Bit127:64ImmSourceExists If:Format:127:64RegSourceExists If:Format:63:32Operand CorFormat:Format:	


dp2 - Dot Product 2				
31:0	Header	Header		
	Format:	EU_INSTRUCTION_HEADER		



dp3 - Dot Product 3

Source: EuIsa

Length Bias:

The dp3 instruction performs a three-wide dot product on four-tuple vector basis and storing the same scalar result per four tuple to all four channels in dst. This instruction is similar to dp4 except that every fourth element of src0 (post-source-swizzle if present) is not involved in the computation.

The dot product of two vectors of equal length is the sum of the products of each pair of corresponding elements.

The dp4 instruction includes all four elements of each vector in the dot product. The dp2 instruction includes the first two elements of each vector in the dot product.

Format:

[(pred)] dp3[.cmod] (exec_size) dst src0 src1

4

Restriction

Restriction: Execution size cannot be less than 4.

Restriction: Horizontal strides must be 1.

Restriction: Source operands cannot be accumulators.

Syntax

[(pred)] dp3[.cmod] (exec_size) reg reg reg [(pred)] dp3[.cmod] (exec_size) reg reg imm32

Pseudocode

Evaluate(WrEn); for (n = 0; n < exec_size; n += 4) { fTmp = src0.chan[n] * src1.chan[n]
+ src0.chan[n+1] * src1.chan[n+1] + src0.chan[n+2] * src1.chan[n+2]; if (WrEn.chan[n])
dst.chan[n] = fTmp; if (WrEn.chan[n+1]) dst.chan[n+1] = fTmp; if (WrEn.chan[n+2])
dst.chan[n+2] = fTmp; if (WrEn.chan[n+3]) dst.chan[n+3] = fTmp; }</pre>

Predication	n Condition	al Modifier	Modifier Saturation Source Modifier				
Υ	Y		Y Y				
Src Types	Dst Types						
F	F						
DWord	Bit		Description				
03	127:64	ImmSour	ImmSource				
		Exists If:	([Imm	Source][Src1.RegF	le]=='IMM')		
		Format:	EU_IN	ISTRUCTION_SOUF	CES_REG_IMM		
	127:64	RegSour	ce				
		Exists If:	([Reg	Source][Src1.RegFi	e]!='IMM')		
		Format:	EU_IN	NSTRUCTION_SOUF	RCES_REG_REG		
	63:32	Operand	Controls				
		Format:	EU IN	STRUCTION OPERA	ND CONTROLS		



dp3 - Dot Product 3				
31:0	Header	Header		
	Format:	EU_INSTRUCTION_HEADER		



dp4 - Dot Product 4

Source: EuIsa

Length Bias:

The dp4 instruction performs a four-wide dot product on four-tuple vector basis and storing the same scalar result per four tuple to all four channels in dst.

The dot product of two vectors of equal length is the sum of the products of each pair of corresponding elements.

Format:

[(pred)] dp4[.cmod] (exec_size) dst src0 src1

4

Restriction

Restriction: Execution size cannot be less than 4.

Restriction: Horizontal strides must be 1.

Restriction: Source operands cannot be accumulators.

Syntax

[(pred)] dp4[.cmod] (exec_size) reg reg reg [(pred)] dp4[.cmod] (exec_size) reg reg imm32

Pseudocode

Evaluate(WrEn); for (n = 0; n < exec_size; n += 4) { fTmp = src0.chan[n] * src1.chan[n] + src0.chan[n+1] * src1.chan[n+1] + src0.chan[n+2] * src1.chan[n+2] + src0.chan[n+3] * src1.chan[n+3]; if (WrEn.chan[n]) dst.chan[n] = fTmp; if (WrEn.chan[n+1]) dst.chan[n+1] = fTmp; if (WrEn.chan[n+2]) dst.chan[n+2] = fTmp; if (WrEn.chan[n+3]) dst.chan[n+3] = fTmp; }

Predication	Conditional Modifier	Saturation	Source Modifier
Y	Υ	Y	Y

Src Types Dst Types

F					
DWord	Bit		Description		
03	127:64	ImmSource			
		Exists If:	([ImmSource][Src1.RegFile]=='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_REG_IMM		
	127:64	RegSource	RegSource		
		Exists If:	Exists If: ([RegSource][Src1.RegFile]!='IMM')		
		Format:	Format: EU_INSTRUCTION_SOURCES_REG_REG		
	63:32	Operand Con	Operand Controls		
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
	31:0	Header	Header		
		Format:	EU_INSTRUCTION_HEADER		



dph - Dot Product Homogeneous

Source:

Length Bias:

The dph instruction performs a four-wide homogeneous dot product on four-tuple vector basis and storing the same scalar result per four tuple to all four channels in dst. This instruction is similar to dp4 except that every fourth element of src0 (post-source-swizzle if present) is forced to 1.0f.

Use the dp4 instruction to do a four-wide dot product that includes all elements of src0 and src1.

Format:

[(pred)] dph[.cmod] (exec_size) dst src0 src1

EuIsa

4

Restriction

Restriction: Execution size cannot be less than 4.

Restriction: Horizontal strides must be 1.

Restriction: Source operands cannot be accumulators.

Syntax

[(pred)] dph[.cmod] (exec_size) reg reg reg [(pred)] dph[.cmod] (exec_size) reg reg imm32

Pseudocode

Evaluate(WrEn); for (n = 0; n < exec_size; n += 4) { fTmp = src0.chan[n] * src1.chan[n] + src0.chan[n+1] * src1.chan[n+1] + src0.chan[n+2] * src1.chan[n+2] + src1.chan[n+3]; // Use 1.0f in place of src0.chan[n+3]. if (WrEn.chan[n]) dst.chan[n] = fTmp; if (WrEn.chan[n+1]) dst.chan[n+1] = fTmp; if (WrEn.chan[n+2]) dst.chan[n+2] = fTmp; if (WrEn.chan[n+3]) dst.chan[n+3] = fTmp; }

Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Υ

Src Types Dst Types

F F					
DWord	Bit		Description		
03	127:64	ImmSource			
		Exists If:	([ImmSource][Src1.RegFile]=='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_REG_IMM		
	127:64	RegSource	RegSource		
		Exists If:	([RegSource][Src1.RegFile]!='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_REG_REG		
	63:32	Operand Con	ntrols		
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
	31:0	Header			
		Format:	EU_INSTRUCTION_HEADER		



	е	lse - Else				
Source: Eu	ılsa					
Length Bias: 4						
The else instruction is a the else/endif portion t inactive before entering	The else instruction is an optional statement within an if/else/endif block of code. It restricts execution within the else/endif portion to the opposite set of channels enabled under the if/else portion. Channels which were inactive before entering the if/endif block remain inactive throughout the entire block.					
All enabled channels u are redirected (by else target should be the th	pon arriving at the else in or before else), a relative j e matching endif instructi	struction are redired ump is performed t on for that conditio	cted to the matching endif. If all channels o the location specified by JIP. The jump nal block.			
The following table de (signed word integer). J forward referencing. Th	scribes the 16-bit JIP. In G IP must be an immediate is value is added to IP pre	EN binary, JIP is at l operand, it is a sigr e-increment.	ocation src1 and must be of type W ned 16-bit number and is intended to be			
Format: else (exec_size) JIP						
		Restriction				
Restriction: Predication	is not allowed.					
Restriction: The executi	on size must be the same	for the if, else, and	endif instructions of the same code block.			
		Suntau				
olao (oroa aizo) im	m16	Syntax				
		Pseudocode				
<pre>Evaluate(WrEn); for } } if (PcIP != (I</pre>	r (n = 0; n < 32; n++ P + 1)) { // for all	-) { if (WrEn.c L channels Jump(]	channel[n]) { PcIP[n] = IP + JIP; IP + JIP); }			
Predication Condition	nal Modifier Saturation	Source Modifier				
N N	Ν	Ν				
DWord Bit		Descr	iption			
03 127:112	UIP					
	Format:		S15			
	The jump distance in nur	nber of eight-byte i	units if a jump is taken for the channel.			
111:96	JIP					
	Format:		S15			
	The jump distance in nur	mber of eight-byte u	units if a jump is taken for the instruction.			
95.64	Reserved					
55.04	Format:		MBZ			
63:32	Operand Control					



else - Else					
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS			
31:0	Header				
	Format:	EU_INSTRUCTION_HEADER			



			en	dif - End I	f		
Source:		EuIsa					
Length E	Length Bias: 4						
			Des	scription			
The end that we	dif instru re active	ction terminates an in prior to the if/else/e	f/else/endif k ndif block.	block of code. It res	stores execution to	the channels	
The en outer ce	dif instru ondition	iction is also used to al block when all cha	hop out of r nnels are dis	nested conditionals abled.	by jumping to the	end of the next	
The foll W (sign added 1	owing ta led word to IP pre	ble describes the 16 integer). JIP must be -increment.	-bit JIP. In GI an immedia	EN binary, JIP is at l ate operand, it is a s	ocation src1 and m signed 16-bit numb	nust be of type per. This value is	
Format: endif J	: IP						
			Re	striction			
Restrict	ion: Pred	lication is not allowe	d.				
Restrict code bl	ion: The ock.	execution size must	be the same	for the if, else, and	endif instructions	of the same	
			Syntax				
endif	(exec_s	ize) imm16					
				Pseudocode			
Evalua	te(WrEr); if (WrEn == 0)) { // al	l channels fals	e Jump(IP + JIP); }	
Predica	ation Co	onditional Modifier	Saturation	Source Modifier			
N	N		N	N			
DWord	Bit			Descript	ion		
03	127:112	Reserved					
		Format:			MBZ		
	111:96	JIP					
	Format: S15						
		Jump Target Offset. The relative offset in 64-bit units if a jump is taken for the instruction.					
	95:91	Reserved					
		Format:			MBZ		
	90	Flag Register Nun Added a second fla	n ber ag register				
	89	Flag Subregister N This field specifies	lumber the sub-regi	ster number for a fl	lag register operan	d. There are two sub-	



endif - End If

	registers ir The select instruction for the inst conditiona	egisters in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the nstruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.				
88:64	Source 0					
	Exists If:	(Stru	cture[l	EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align1')		
	Format:	EU_I	NSTRU	CTION_OPERAND_SRC_REG_ALIGN1		
88:64	Source 0					
	Exists If:	(Stru	cture[E	EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align16')		
	Format:	EU_IN	ISTRU	CTION_OPERAND_SRC_REG_ALIGN16		
63:32	Operand 0	Contro	bl			
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS				
31:0	Header					
	Format:			EU_INSTRUCTION_HEADER		



math - Extended Math Function

Source: EuIsa

Length Bias:

The math instruction performs extended math function on the components in src0, or src0 and src1, and write the output to the channels of dst. The type of extended math function are based on the FC[3:0] encoding in the table below.

Format:

[(pred)] math (exec_size) dst src0 src1 <FC>

4

Restriction

Restriction: Accumulator access is allowed only for ieee macro functions.

Restriction: The math instruction does not support indirect addressing modes.

Restriction: The only supported rounding mode for math instruction is Round to Nearest Even.

Restriction: INT DIV function does not support SIMD16.

Restriction: The FDIV function is not supported in ALT_MODE.

Restriction: The math instruction must use GRF registers as source(s) and destination.

Restriction: The supported regioning mode for math instruction is align1 with the following restrictions:

Scalar source is supported.

Source and destination horizontal stride must be 1.

Width must be the same as execution size.

Source and destination offset must be the same, except the case of scalar source.

Syntax

[(pred)] math (exec_size) reg reg reg imm4

Pseudocode

```
Evaluate(WrEn);
for (n = 0; n < exec_size; n++) {</pre>
    if (WrEn.channel[n] == 1) {
        switch FC[3:0] {
            case 1h:
                dst.channel[n] = rcp(src0.channel[n]);
            case 2h:
                dst.channel[n] = log(src0.channel[n]);
            case 3h:
                dst.channel[n] = exp(src0.channel[n]);
            case 4h:
                dst.channel[n] = sqrt(src0.channel[n]);
            case 5h:
                dst.channel[n] = rsq(src0.channel[n]);
            case 6h:
                dst.channel[n] = sin(src0.channel[n]);
            case 7h:
                dst.channel[n] = cos(src0.channel[n]);
            case 9h: // src0 / src1
                dst.channel[n] = fdiv(src0.channel[n], src1.channel[n]);
            case Ah:
               dst.channel[n] = pow(src0.channel[n], src1/channel[n]);
            case Bh: // src0 / src1
```



math - Extended Math Function

```
idiv(src0.channel[n], src1.channel[n]);
dst.channel[n] = quotient;
dst+1.channel[n] = remainder;
case Ch:
    idiv(src0.channel[n], src1.channel[n]);
    dst.channel[n] = quotient;
case Dh:
    idiv(src0.channel[n], src1.channel[n]);
    dst.channel[n] = remainder;
}
```

Predication	Conditional Modifier	Saturation	Source Modifier
Y	Ν	Y	Ν

Src TypesDst TypesFFDDUDUD

}

}

DWord	Bit	Description						
03	127:64	RegSource						
		Format:	Format: EU_INSTRUCTION_SOURCES_REG_REG					
	63:32	Operand Control						
		Format:	Format: EU_INSTRUCTION_OPERAND_CONTROLS					
	31:28	Controls B	Controls B					
		Format:	Format: EU_INSTRUCTION_CONTROLS_B					
	27:24	Function Control (FC)						
		Format: FC						
	23:8	Controls A						
		Format:	EU_INSTRUCTI	ON_CONTROLS_A	١			
	7	Reserved						
		Format:	Format: MBZ					
	6:0	Opcode	Opcode					
		Format:		EU_OPCODE				



fbl - Find First Bit from LSB Side

Source:

Length Bias:

The fbl instruction counts component-wise the number of LSB 0 bits before the first 1 bit in src0, storing that number in dst.

Format:

[(pred)] fbl (exec_size) dst src0

Programming Notes

If src0 contains no 1 bits, store 0xFFFFFFFF in dst.

EuIsa

4

Restriction

Restriction: No accumulator access, implicit or explicit.

Syntax

[(pred)] fbl (exec_size) reg reg [(pred)] fbl (exec_size) reg imm32

Pseudocode

$Evaluate(WrEn);$ for (n = 0; n < exec_size; n++) { if ($WrEn.chan[n]$) { UD cnt = 0; UD
udScalar = src0.chan[n]; while ((udScalar & 1) == 0 && cnt != 32) { cnt ++; udScalar =
udScalar >> 1; } if (src0.chan[n] == 0x00000000) { dst.chan[n] = 0xFFFFFFF; } else {
dst.chan[n] = cnt; } } }

Predication	Conditional Modifier	Saturation	Source Modifier
Y	Ν	Ν	Ν

Src Types Dst Types

	D							
DWord	Bit	Description						
03	127:64	ImmSource						
		Exists If:	([Operand Controls][Src0.RegFile]=='IMM')					
		Format:	EU_INSTRUCTION_SOURCES_IMM32					
	127:64	RegSource						
		Exists If:	([Operand Controls][Src0.RegFile]!='IMM')					
		Format:	EU_INSTRUCTION_SOURCES_REG					
	63:32	Operand Cont	trols					
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS					
	31:0	Header						
		Format:	EU_INSTRUCTION_HEADER					



fbh - Find First Bit from MSB Side

Source:	EuIsa	a							
Length Bias:	4								
If src0 is unsi- resulting cou	If src0 is unsigned, the fbh instruction counts component-wise the leading zeros from src0 and stores the resulting counts in dst.								
If src0 is sigr stores the res	If src0 is signed and positive, the fbh instruction counts component-wise the leading zeros from src0 and stores the resulting counts in dst.								
If src0 is sigr stores the res	If src0 is signed and negative, the fbh instruction counts component-wise the leading ones from src0 and stores the resulting counts in dst.								
Format: [(pred)] fbh ((exec_size) ds	st src0							
			Pro	gramming Notes					
If src0 is zero	, store 0xFFF	FFFFF in dst.							
If src0 is sign	ed and is -1 ((OxFFFFFFF)	, store 0xFF	FFFFFF in dst.					
				Restriction					
Restriction: N	lo accumulat	or access, in	plicit or ex	plicit.					
				Syntax					
[(pred)] fk	oh (exec si	ze) reg re	a [(pred)	l fbh (exec siz	e) reg imm32				
[(]]]	/// (Cheo_c _	20, 105	.9 ((Free),] 1011 (01100_22-					
				Pseudocode					
<pre>Evaluate(Wr src0 is uns cnt != 32) dst.chan[n] dScalar = s cnt != 32) (src0.chan[} }</pre>	<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { UD cnt = 0; if (src0 is unsigned) { UD udScalar = src0.chan[n]; while ((udScalar & (1 << 31)) == 0 && cnt != 32) { cnt ++; udScalar = udScalar << 1; } if (src0.chan[n] == 0x00000000) { dst.chan[n] = 0xFFFFFFF; } else { dst.chan[n] = cnt; } } else { // src0 is signed. D dScalar = src0.chan[n]; bit cval = dScalar[31]; while ((dScalar & (1 << 31)) == cval && cnt != 32) { cnt ++; dScalar = dScalar << 1; } if (src0.chan[n] == 0xFFFFFFF;) (src0.chan[n] == 0x00000000) { dst.chan[n] = 0xFFFFFFF; } else { dst.chan[n] = cnt; } } }</pre>								
Predication	Conditional	Modifier	Saturation	Source Modifier					
Υ	Ν		Ν	Ν					
Src Types D	st Types								
D,UD U	D								
DWord	Bit	Description							
03	127:64	ImmSourc	e						
		Exists If:	([Opera	and Controls][Src0.	RegFile]=='IMM')				
		Format:	EU_INS	STRUCTION_SOUR	CES_IMM32				
	127:64	RegSource							
		Exists If:	([Oper	and Controls][Src0	.RegFile]!='IMM')				



fbh - Find First Bit from MSB Side							
		Format:	EU_I	NSTRUCTION_SOURCES_REG			
	63:32	Operand Cont	rols	ols			
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS				
31:0 Header							
		Format:		EU_INSTRUCTION_HEADER			



frc - Fraction							
Source:	EuIs	а					
Length Bias:	4						
The frc instruction computes, component-wise, the truncate-to-minus-infinity fractional values of src0 and stores the results in dst. The results, in the range of [0.0, 1.0], are the fractional portion of the source data. The result is in the range [0.0, 1.0] irrespective of the rounding mode. Floating-point fraction computation follows the rules in the following tables, based on the current floating-point mode.							
Format: [(pred)] frc[.c	cmod] (exec_	size) dst src()				
				Syntax			
[(pred)] fr	cc[.cmod]	(exec_size)) reg reg	g [(pred)] frc[.cmod] (exec_size) reg imm32			
				Pseudocode			
Evaluate(Wr src0.chan[r	rEn); for n] - floor	(n = 0; n (src0.chan)	< exec_s [n]);	size; n++) { if (WrEn.chan[n]) { dst.chan[n] = }			
Predication Y	Condition a	l Modifier	<mark>Saturatio</mark> N	on Source Modifier			
Src TypesDFF	st Types	I					
DWord	Bit			Description			
03	127:64	ImmSourc	e				
		Exists If:	([Ope				
	Format: EU_INSTRUCTION_SOURCES_IMM32						
	127:64 RegSource						
	Exists II. ([Operand Controls][SICO.Regrine]:= INNN)						
	63.32	Concreated Controls					
	05.52	Format:	EU_IN	NSTRUCTION_OPERAND_CONTROLS			
	31:0	Header	I				
		Format:		EU_INSTRUCTION_HEADER			



				GF	PGP	U_(OBJEC	СТ			
Source:		Re	ende	rCS							
Length E	Bias:	2									
DWord	Bit						Description	on			
0	31:29	Comma	nd T	уре			-				
		Default	Valu	ie:				3h GFXPIPE			
		Format:						OpC	Code		
	28:27	Pipeline									
		Default	Valu	ie:					2h Media		
		Format:							OpCode		
	26:24	Media C	omi	mand Opcode							
		Default	Valu	ie:			1h GPGPL	J_OBJI	ECT		
		Format:					OpCode				
	23:16	SubOpc	ode								
		Default	Valu	ie:	04	h GPC	GPU_OBJE	CT Suł	bOp		
		Format:			Op	oCode	<u>e</u>				
	15:9	Reserve	d						-		
		Format:							MBZ		
	8	Predicat	te Er	nable							
		Format:						Enable	e		
		If set, thi internal s bit is 0.	s co state	mmand is execute bit. This commar	ed (or r nd is ig	not) d norec	epending 1 only if Pr	on the edicat	e current value of the MI Predicate teEnable is set and the Predicate state		
	7:0	DWord Length									
		Format: =n Total Length -2									
		There are	e 4 [DW needed to spe	ecify the	e Thre	ead Group	ad Group ID and the execution mask.			
		Value	•		Nar	me			Description		
		6h		DWORD_COUNT_	n [Def	ault]			Excludes DWord (0,1)		
1	31:8	Reserve	d								
	7	Shared I This bit, group is	Shared Local Memory Fixed Offset This bit, if set, specifies that the offset into the 64k Shared Local Memory for the current thread group is specified by software in the Shared Local Memory Offset field								
		Value		Name				[Description		
		0	Thre Offse	ad Groups et	Offset concui	to sta rrently	art of segment determined by hardware based on y running thread groups.				
			Shar Men	ed Local hory Offset	Offset Sharec	to sta d Loca	art of the S al Memory	Shared Offse	d Local Memory segment supplied in et		
	6:5	Reserve	d								
		Format: MBZ							MBZ		



		G	PGPU_OBJECT						
	4:0	Interface Descriptor Offset							
		Format:		U5					
		This field specifies the offset from the interface descriptor base pointer to the interface descriptor which will be applied to this object. It is specified in units of interface descriptors. In VLD mode, this field is ignored by hardware.							
2	31:28	Shared Local Memory Offse	t						
-	51.20	Format:		U4					
		If the Shared Local Memory F Shared Local Memory for this starting address. All threads i	ixed Offset is set, this field pr thread group. The value of t n the thread group must have	ovides the offset to the start of the his field is multiplied by 4k to get the e the same value.					
	27:25	Reserved							
		Format:		MBZ					
	24	End of Thread Group							
		Format:	Boolean						
		This bit indicates that this dispatch is the last for the current thread group.							
	23:19	Reserved							
		Format:		MBZ					
	18:17	Half-Slice Destination Select This field selects the half slice that this thread must be sent to.							
		Value		Name					
		00b	Either half-slice						
		01b Half-Slice 0							
		10b	Half-Slice 1						
	16:0	Indirect Data Length							
		Format:	U17 in bytes						
		This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Data Start Address field is ignored. This field must have the same alignment as the Indirect Object Data Start Address. DWord Length = 0							
3	31:0	Indirect Data Start Address							
		Format: Indirec	tObjectBaseOffset[31:0]						
		This field specifies the Graphi	cs Memory starting address o	of the data to be loaded into the kernel					
		for processing. This pointer is	relative to the Indirect Obje	ct Base Address.					
		The start address is DWord al (Bits 31:29 MBZ)	igned address.						
		Val	ue	Name					
		[0,512MB]							



GPGPU_OBJECT							
4	31:0	Thread Group ID X This is the X coordinate of the group id.					
5	31:0	Thread Group ID Y This is the Y coordinate of the group id for all channels generated by this command.					
6	31:0	Thread Group ID Z This is the Z coordinate of the thread group id.					
7	31:0	Execution Mask					
		Format:	Must Be All Ones Must be 0xFFFFFFF				
		per channel enable for the SIMD32 dispatch. The LSB of the Mask enables the 32 channel 0; the remaining bits enable the corresponding channel numbers. 8 dispatches should use the LSB bits of the mask. Any disabled channel will not to memory.					



			GPG	PU_\	WALK	ER			
Source:		Rende	rCS						
Length E	Bias:	2							
DWord	Bit	Description							
0	31:29	Command T	уре					1	
		Default Valu	ie:			3h GF	FXPIPE	<u> </u>	
		Format:				ОрСо	ode		
	28:27	Pipeline				ľ			
		Default Valu	ie:			2	h Meo	dia	
		Format:				С	DpCod	le	
	26:24	Media Com	mand Opcode						
		Default Valu	Default Value: 1h GPGP			_WALK	ER		
		Format:			OpCode				
	23:16	SubOpcode	Α						
		Default Valu	ie:	05h GP	GPU_WALK	ER Sub	Ор		
		Format:		OpCod	e				
	15:11	Reserved							
		Format:					MBZ		
	10	Indirect Para	ameter Enable		T				
		Format:				Enable			
		If set, the values in DW 4, 6, 8 are ignored and replaced by the current values of the corresponding GPGPU_xxx MMIO registers: GPGPU_DISPATCHDIMX (instead of DW4) GPGPU_DISPATCHDIMY (instead of DW6) GPGPU_DISPATCHDIMZ (instead of DW8)						irrent values of the	
	9	Reserved							
		Format:				MBZ			
	8	Predicate Er	nable						
		Format:				Enable			
		If set, this command is executed (or not) depending on the current value of the MI Predicate internal state bit. This command is ignored only if PredicateEnable is set and the Predicate state bit is 0.							
	7:0	DWord Leng	gth						
		Format:						=n	
		Total Length	- 2						
		Value		Name	9			Description	
		9h	DWORD_COUNT_n	[Defaul	t]			Allowed value is 9	
1	31:8	Reserved							

Г



			GI	PGPU_WALKER					
	7:5	Reserved							
		Format:			MBZ				
	4:0	Interface De	escriptor Offset						
		Format:			U5				
		This field spe	ecifies the offset	from the interface descriptor	base pointer to the interface descriptor				
		which will be	e applied to this	object. It is specified in units of	of interface descriptors.				
2	31:30	SIMD Size This field determines the size of the payload and the number of bits of the execution mask that are expected. The kernel pointed to by the interface descriptor should match the SIMD declared here.							
		Value	Name		Description				
		0	SIMD8	8 LSBs of the execution mas	<pre>< are used</pre>				
		1	SIMD16	16 LSBs used in execution m	ask				
		2	SIMD32	32 bits of execution mask us	ed				
	29:22	Reserved							
		Format:			INIBZ				
		The maximum value + 1. (T equal Numb	m value of the th hread_Depth_N per of Threads in	nread depth counter. Since the Max+1)*(Thread_Height_Max n GPGPU Thread Group in th	e counter starts at 0, the depth is this x+1)*(Thread_Width_Max+1) must he Interface Descriptor.				
	15:14	Reserved							
		Format:			MBZ				
	13:8	Thread Heig The maximu	ght Counter Ma m value of the th	ximum nread height counter. The heig	ght is this value + 1.				
	7:6	Reserved							
		Format:			MBZ				
	5:0	Thread Wid The maximu	th Counter Max m value of the th	kimum hread width counter. The heig	ht is this value + 1.				
3	31:0	Thread Group ID Starting X This is the initial value of the X component of the thread group. When X reaches the maximum value it rolls around to 0, not to this value.							
4	31:0	Thread Group ID X Dimension The X dimension of the thread group (maximum X is dimension -1)							
5	31:0	Thread Grou This is the in value it rolls	up ID Starting Y itial value of the around to 0, not	Y component of the thread g t to this value.	roup. When Y reaches the maximum				
6	31:0	Thread Grou The Y dimen	u p ID Y Dimens sion o <mark>f the threa</mark>	ion ad group (maximum Y is dime	nsion -1)				
7	31:0	Thread Grou This is the in	up ID Starting Z itial value of the	Z Z component of the thread g	roup				



	GPGPU_WALKER						
8	31:0	Thread Group ID The Z dimension	7 Dimension of the thread group (maximum Z is dimension -1)				
9	31:0	Right Execution	Mask				
		Format:	Must Be All Ones Must be 0xFFFFFFF				
			Programming Notes				
		When simulating execution masks	SIMD64 to fit large thread groups this field must be 0xFFFFFFFF and the actual passed in the payload.				
10	31:0	Bottom Executio	n Mask				
		Format:	Must Be All Ones Must be 0xFFFFFFF				
		Programming Notes					
		When simulating execution masks	SIMD64 to fit large thread groups this field must be 0xFFFFFFFF and the actual passed in the payload.				



f16to32 - Half Precision Float to Single Precision Float

Source:

Length Bias:

The f16to32 instruction converts the half precision float in src0 to single precision float and storing in dst.

Because this instruction does not have a 16-bit floating-point type, the source data type must be Word (W). The destination type must be F (Float).

Format:

[(pred)] f16to32[.cmod] (exec_size) dst src0

EuIsa

4

Restriction

Restriction: The FP Mode (Single Precision Floating Point Mode in cr0) must be IEEE mode.

Restriction: No accumulator access, implicit or explicit.

Syntax

[(pred)] f16to32[.cmod] (exec_size) reg reg [(pred)] f16to32[.cmod] (exec_size) reg imm16

				Pseudocode			
Evaluate(W convert ha	<pre>valuate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = convert half precision float to single precision float(src0.chan[n]); } }</pre>						
Predication	Conditiona	al Modifier	Saturation	Source Modifier			
Y	Y		Y	Υ			
Src Types	Src Types Dst Types						
W F	:						
DWord	Bit			Description			
03	127:64	ImmSour	ce				
		Exists If:	([Opera	rand Controls][Src0.RegFile]=='IMM')			
		Format:	EU_INS	STRUCTION_SOURCES_IMM32			
	127:64	RegSourc	e				
		Exists If:	([Oper	rand Controls][Src0.RegFile]!='IMM')			
		Format:	EU_INS	ISTRUCTION_SOURCES_REG			
	63:32	Operand	Operand Controls				
		Format:	Format: EU_INSTRUCTION_OPERAND_CONTROLS				
	31:0	Header					
		Format:	E	EU_INSTRUCTION_HEADER			



	halt - Halt									
Source:	Source: EuIsa									
Length Bias	Length Bias: 4									
			Des	cription						
The halt in execution, HALT, jump	struction ten the enabled o to the insti	nporarily susp channels are ruction at (IP +	ends executi sent to the iı + JIP).	on for all enabled on for all enabled on for all enabled on the second sec	compute channels. Upon UIP), if all channels are enabled	d at				
If the halt same. If the program a	instruction is e halt instruc nd the JIP sh	s not inside ar ction is inside ould be the e	ny conditiona a conditiona nd of the inr	al code block, the v l code block, the U ner most conditiona	values of JIP and UIP should be IP should be the end of the al code block.	e the				
The UIP m	ust point to	a HALT Instru	ction.							
If SPF is O	N, the UIP m	nust be used to	o update IP;	JIP is not used in th	his case.					
The followi signed 16- must be of	ing table des bit numbers type W (sig	scribes the two , added to IP I ned word inte	o 16-bit instr ore-incremer ger).	uction pointer offs nt. In GEN binary, JI	ets. Both the JIP and UIP are IP and UIP are at location src1	and				
Format: [(pred)] ha	llt (exec_size) JIP UIP								
				Restriction						
Restriction	: dst and src	0 must be NU	LL.							
[(pred)]	halt (exec	size) imm1	Synta	X						
r (Frod) 1										
				Pseudocode						
Evaluate(else { Pc }	WrEn); for :IP[n] = IE	r (n = 0; n p + 1; } } i	n < 32; n++ .f (PcIP !) { if (WrEn. = (IP + 1)) {	<pre>channel[n]) { PcIP[n] = // for all channels Jump(</pre>	IP + UIP; (IP + JIP);				
Predicatio	n Conditio	nal Modifier	Saturation	Source Modifier						
Y	Ν		Ν	Ν						
DWord	Bit			Descr	ription					
03	127:112	UIP				1				
		The jump dis	Format:S15The jump distance in number of eight-byte units if a jump is taken for the channel.							
	111:96	JIP								
		Format:			S15					
		The jump dis	tance in nun	nber of eight-byte	units if a jump is taken for the	instruction.				



halt - Halt								
95:64	Reserved	Reserved						
	Format:			MBZ				
63:32	Operand Cont	rol						
	Format:	EU_IN	ISTRUCTION_OPERAND_CONTR	ROLS				
31:0	Header							
	Format:		EU_INSTRUCTION_HEADER					



	if - If										
Source:	E	uIsa	a								
Length Bias	: 4										
	Description										
An if instru conditiona	ction starts I block to or	an if/endif or a nly those chan	an if/else/en nels that wei	dif block of code. I re enabled via the j	It restricts execution within the predicate control.						
Each if instruction	truction mus before the r	st have a matc matching endi	hing endif ir f.	nstruction and may	have up to one matching else						
If all chanr instruction present, or	nels are inac referenced otherwise to	tive (for the if, by JIP. This jur o the matchin	/endif or if/e np must be g endif instru	lse/endif block), a j to right after the m uction of the condi	jump is performed to the natching else instruction when itional block.						
If SPF is O	N, the UIP m	nust be used to	o update IP;	JIP is not used in t	his case.						
The followi signed 16- must be of	ing table des bit numbers type W (sig	scribes the two , added to IP r ned word inte	o 16-bit instr ore-incremer ger).	ruction pointer offs nt. In GEN binary, J	sets. Both the JIP and UIP are IP and UIP are at location src1 and						
Format:											
[(pred)] if	(exec_size) J	IP UIP									
				Restriction							
Restriction	: The execut	ion size must	be the same	for the if, else, and	d endif instructions of the same code	block.					
			Synta	x							
[(pred)]	if (exec_s	size) imm16	imm16								
				Pseudocode							
Evaluate(JIP; } el + JIP); }	WrEn); for se { PcIP	r (n = 0; n [n] = IP + 1	<pre>1 < 32; n++ ; } } if (</pre>	-) { if (WrEn. PcIP != (IP +	<pre>cchannel[n] == 0) { PcIP[n] = 1)) { // for all channels Jur</pre>	IP + np(IP					
Predicatio	n Conditio	nal Modifier	Saturation	Source Modifier							
Υ	Y		Ν	Ν							
DWord	Bit			Desci	ription						
03	127:112	UIP									
		Format: S15									
		The jump dis	tance in nun	nber of eight-byte	units if a jump is taken for the chann	el.					
	111:96	JIP									
		Format:			S15						
		The jump dis	tance in nun	nber of eight-byte	units if a jump is taken for the instru-	ction.					



if - If								
95:64	Reserved	Reserved						
	Format:			MBZ				
63:32	Operand Cont	rol						
	Format:	EU_II	NSTRUCTION_OPERAND_CONTR	ROLS				
31:0	Header							
	Format:		EU_INSTRUCTION_HEADER					



		i	lleg	gal - Illega	d -	
Source:	EuI	sa				
Length Bias:	4					
The Illegal Op is to transfer	ocode Excer control to t	otion Enable flag in he System Routine	cr0.1	is normally set so t	the normal proce	essing of an illegal opcode
Instruction d opcode expan	ispatch trea nsion) as if	ats any unused 8-b it is the illegal opco	it opco ode.	ode (including bit 7	' of the instructio	on, reserved for future
The illegal of instruction po	ocode is zei pinter.	ro because that by	e valu	ie is more likely tha	in most to be rea	id via a wayward
The illegal in are special va	struction is lues indicat	an instruction only ting invalid instance	r in the es.	e same way that a l	NULL pointer in s	oftware is a pointer. Both
Format: illegal						
				Restriction		
Restriction: T	he illegal in	struction takes no	instru	ction options.		
				Syntax		
illegal						
				Pseudocode		
{ Set the I Enable is s IP = SIP).	illegal Op set in cr0 } }	code Exception 0.1) { Transfer	Statu cont	s bit in cr0.1. rol to the Syst	if (Illegal em Routine (re	Opcode Exception eturn address to AIP,
Predication	Condition	al Modifier Satur	ation	Source Modifier		
N	N	N		N		
		· · · · · · · · · · · · · · · · · · ·				
DWo	rd	Bit			Description	
03		127:7	Rese	erved		1
			Forr	mat:		MBZ
		6:0	Орсо	ode		
Format: EU_OPCODE						



	sub	b - Intege	er S	ubtractior	n with Borrow	
Source:	EuIs	а				
Length Bias:	4					
The subb inst stores the bo	ruction perf rrow into ac	orms component c.	-wise s	subtraction of src0 a	and src1 and stores the results in dst, it also	
If the operati	ion produce	s a borrow (src0 ·	< src1),	, write 0x0000001	to acc, else write 0x00000000 to acc.	
Format: [(pred)] subb	[.cmod] (exe	ec_size) dst src0 s	rc1			
				Restriction		
Restriction: A destination o	ccWrEn is re perand.	equired. The accu	mulato	or is an implicit dest	ination and thus cannot be an explicit	
				Syntax		
[(pred)] su imm32	lbb[.cmod]	(exec_size) re	eg reg	g reg [(pred)] s	ubb[.cmod] (exec_size) reg reg	
				D		
Proglam to (Mar		(Pseudocode		
src0.chan[n	En); for] - src1.0	(n = 0; n < e) chan[n]; acc.ch	nan[n]	= borrow(src.cl	(wrEn.chan[n]) { dst.chan[n] = han[n] - src1.chan[n]); } }	
Predication	Conditiona	l Modifier Satu	ration	Source Modifier		
Y	N	Y		Ν		
Src Types D	st Types					
UD UI	D					
DWord	Bit			Dese	cription	
03	127:64	ImmSource	-			
		Exists If:	([Imn	([ImmSource][Src1.RegFile]=='IMM')		
		Format:	EU_IN	NSTRUCTION_SOUR	RCES_REG_IMM	
	127:64	RegSource	1			
		Exists If: ([RegSource][Src1.RegFile]!='IMM')				
		Format: EU_INSTRUCTION_SOURCES_REG_REG				
	63:32	Operand Cont	rols			
		Format:	EU_IN	STRUCTION_OPERA	ND_CONTROLS	
	31:0	Header				
		Format:	E	EU_INSTRUCTION_H	IEADER	



jmpi - Jump Indexed									
Source: EuIsa									
Length Bias: 4									
			Des	scription					
The jmpi inst instruction p jumps, and n	ruction red ointer. The egative inte	lirects progra index is a sig egers for bac	m execution ned integer kward jumps	to an index offset relative to the post-incremented value, with positive or zero integers for forward s.					
Note: Unlike instruction p	other flow ointer rathe	control instr er than the IP	uctions, the value for the	offset used by jmpi is relative to the incremented e instruction itself.					
In GEN bina the dst and s	ry, index is rc0 locatio	at location sr ns.	c1. The ip re	gister must be put (for example, by the assembler) at					
Predication is allowed to provide conditional jump with a scalar condition. As the execution size is 1, the first channel of PMASK (flags post prediction control and negate) is used to determine whether the jump is taken or not. If the condition is false, the jump is not taken and execution continues with the next instruction.									
Format: [(pred)] jmpi (1) index {NoMask}									
			Progran	nming Notes					
An index of () does noth	ning, continui	ng executior	n with the next instruction.					
An index of - format) is an	2 (if the jm infinite loo	pi instruction p on the jmp	is in native i instruction	format) or -1 (if the jmpi instruction is in compact					
				Restriction					
Restriction: T	he executio	on size must l	oe 1.						
Restriction: T	he {NoMas	k} instruction	option mus	st be specified.					
Restriction: T	he index da	ata type must	t be D (Signe	ed DWord Integer).					
				Constant					
			1-) [(Syntax					
[(pred)]]	црт (т) те	egsz (Nomas	r} [(pred)) Jupi (I) Iuuusz [Nomask]					
				Pseudocode					
Evaluate(Wi for the IP	cEn); if of the fo	(WrEn != 0 ollowing in) { Jump(struction.	(IP + 1 + index); // IP + 1 is a pseudocode idiom . }					
Predication	Condition	al Modifier	Saturation	Source Modifier					
Υ	Ν		N	Ν					
Src Ty D	pes								



jmpi - Jump Indexed

DWord	Bit			Description					
03	127:112	Reserved							
		Format:			MBZ				
	111:96	JIP							
		Format:			S15				
		Jump Targe	et Offset. Th	ne relative offset in 64-bit units if a j	jump is taken for the instruction.				
	95:91	Reserved							
		Format:			MBZ				
	90	Flag Regis Added a se	ter Numbe	e gister					
		This field specifies the sub-register number for a flag register operand. There are two sub- registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.							
	88:64	Source 0							
		Exists If:	(Structure[[EU_INSTRUCTION_CONTROLS_A][A	<pre>AccessMode]=='Align1')</pre>				
		Format:	EU_INSTRL	JCTION_OPERAND_SRC_REG_ALIGN	V1				
	88:64	Source 0	Source 0						
		Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][A	ccessMode]=='Align16')				
		Format:	EU_INSTRU	JCTION_OPERAND_SRC_REG_ALIGN	J16				
	63:32	Operand C	ontrol						
		Format:	EU_IN	NSTRUCTION_OPERAND_CONTROL	S				
	31:0	Header							
		Format:		EU_INSTRUCTION_HEADER					



		lzd -	Leadi	ng Zero D	etection	
Source:	urce: EuIsa					
Length Bias:	4					
The lzd instr	uction count	s componer	nt-wise the l	eading zeros from s	src0 and stores the resulting counts in dst.	
If src0 is zer	ro, store 32 ii	n dst.				
Format: [(pred)] lzd[.cmod] (exed	_size) dst sr	c0			
				Restriction		
Restriction: A	Accumulator	cannot be c	lestination, i	mplicit or explicit.		
				Syntax		
[(pred)] 1	zd[.cmod]	(exec_size	e) reg reg	[(pred)] lzd[.c	mod] (exec_size) reg reg	
				Pseudocode		
Evaluate(W src0.chan[udScalar =	rEn); for n]; UD cnt udScalar	(n = 0; n = 0; whil << 1; } ds	n < exec_si .e ((udSca st.chan[n]	<pre>lze; n++) { if alar & (1 << 31) = cnt; } }</pre>	(WrEn.chan[n]) { UD udScalar =) == 0 && cnt != 32) { cnt ++;	
Predication	Conditiona	al Modifier	Saturation	Source Modifier		
Y	Y		Y	Y		
Src Types	Ost Types					
D,UD l	JD					
DWord	Bit			Des	cription	
03	127:64	ImmSour	ce			
		Exists If:	([Oper	and Controls][Src0.	RegFile]=='IMM')	
		Format:	EU_INS	STRUCTION_SOUR	CES_IMM32	
	127:64 RegSource					
Exists		Exists If:	([Oper	and Controls][Src0.	RegFile]!='IMM')	
Format:		EU_IN	STRUCTION_SOUR	CES_REG		
	63:32	Operand	Controls			
		Format:	EU_IN:	STRUCTION_OPERA	ND_CONTROLS	
	31:0	Header				
			E	U_INSTRUCTION_H	IEADER	



line - Line

Source: EuIsa

Length Bias:

The line instruction computes a component-wise line equation (v = p * u + q where u, v are vectors and p, q are scalars) of src0 and src1 and stores the results in dst. src1 is the input vector u. src0 provides input scalars p and q, where p is the scalar value based on the region description of src0 and q is the scalar value implied from src0 region. Specifically, q is the fourth component of the 4-tuple (128-bit aligned) that p belongs to.

Format:

[(pred)] line[.cmod] (exec_size) dst src0 src1

4

Restriction

Restriction: This is a specialized instruction that only supports an execution size (ExecSize) of 8 or 16.

Restriction: The src0 region must be a replicated scalar (with HorzStride == VertStride == 0).

Restriction: src0 must specify .0 or .4 as the subregister number, corresponding to a subregister byte offset of 0 or 16.

Restriction: Source operands cannot be accumulators.

Syntax

[(pred)] line[.cmod] (exec_size) reg reg reg [(pred)] line[.cmod] (exec_size) reg reg imm32

Pseudocode

Evaluate(WrEn); for (n = 0; n < exec_size; n++) { dwP = src0.RegNum. SubRegNum[bits4:2]; // A DWord-aligned scalar. dwQ = src0.RegNum.(SubRegNum[bit4] | 0x8); // Fourth component. if (WrEn.chan[n]) { dst.chan[n] = dwP * src1.chan[n] + dwQ; } }

Predication	Conditional Modifier	Saturation	Source Modifier
Y	Υ	Y	Υ

Src Types Dst Types

F F

· ·							
DWord	Bit	Description					
03	127:64	ImmSource					
		Exists If:	Exists If: ([ImmSource][Src1.RegFile]=='IMM')				
		Format: EU_INSTRUCTION_SOURCES_REG_IMM					
	127:64	RegSource					
		Exists If:	xists If: ([RegSource][Src1.RegFile]!='IMM')				
		Format:	ormat: EU_INSTRUCTION_SOURCES_REG_REG				
	63:32	Operand Controls					
Format: EU_INSTRUCTION_OPERAND_CONTROLS				RUCTION_OPERAND_CONTROLS			
	31:0	Header					
		Format:	EU	INSTRUCTION_HEADER			



Irp - Linear Interpolation

Source:

Length Bias:

The lrp instruction takes component-wise multiplication of src0 and src1, and adds the result to the component-wise multiplication of src2 and (1 - src0), and then stores the final results in dst.

Format:

[(pred)] lrp[.cmod] (exec_size) dst src0 src1 src2

EuIsa 4

Restriction

Restriction: The vertical stride (VertStride) is overloaded to 4 in HW for 3-source instructions.

Restriction: The overflow conditional modifier (.o) is not allowed.

Restriction: No explicit accumulator access because this is a three-source instruction. AccWrEn is allowed for implicitly updating the accumulator.

Restriction: All three-source instructions have certain restrictions, described in Instruction Machine Formats.

Syntax

[(pred)] lrp[.cmod] (exec_size) reg reg reg

Pseudocode									
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src1.chan[n] * src0.chan[n] + src2.chan[n] * (1.0 - src0.chan[n]); } }</pre>									
Predica	tion	Con	ditio	nal Modifie	r Saturation	Source Modifier			
Y N		N			Y	Υ			
Src Types Dst T		t Ty	pes						
F F									
DWord	Bit								
03 127:126 Reserved									
			Form	at:		MBZ			
125:106 Source 2									
			Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC						
105			Reserved						
			Format:					MBZ	
	104:8	5	Source 1						
			Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC						
	84 Reserved								
			Format:					MBZ	
	83:64	4	Sourc	e 0					
	Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC							IREE_SRC	



		rp -	Linear Interpolat	ion				
63:56	:56 Destination Register Number							
	Format:		DstRegNum	DstRegNum				
55:53	Destination Su	Destination Subregister Number						
	Format:	Format: DstSubRegNum[2:0]						
52:49	Destination Channel Enable							
	Format:	Format: ChanEn[4]						
	Four channel er	nables a	are defined for controlling which c	hannels are written into the				
	destination regi	ion. The els The	ese channel mask bits are applied are is 1-bit Channel Enable for each	In a modulo-four manner to all channel within the group of 4. If the				
	bit is cleared, th	ne write	for the corresponding channel is	disabled. If the bit is set, the write is				
	enabled. Mnem	ionics f	or the bit being set for the group of	of 4 are <i>x</i> , <i>y</i> , <i>z</i> , and <i>w</i> , respectively,				
	where x corresponds to Channel 0 in the group and w corresponds to channel 3 in the g							
48	Reserved							
	Format:			MBZ				
47	NibCtrl							
	Format: NibCtrl							
46	Reserved							
	Format:			MBZ				
45:44	Destination Da	ta Typ						
	Destination Data Type This field contains the data type for the destination Value Name							
	00h		Single Procision Float	Valle				
	Value Name 00b Single Precision Float 01b DWord							
	10b		Unsigned DWord					
	105 11b		Double Precision Float					
12.10	Source Data T	Inc						
40.42	This field contains the data type for all three sources							
	Value		1	lame				
	00b		Single Precision Float					
	01b		DWord					
	10b		Unsigned DWord					
	11b		Double Precision Float					
41:40	Source 2 Modifier							
	Exists If:	([Pro	operty[Source Modification]=='true')					
	Format:	SrcM	1od					
39:38	Source 1 Modi	Source 1 Modifier						
	Exists If:	([Pro	pperty[Source Modification]=='true	e')				
	Format: SrcMod							



41:36	Reserved						
	Exists If:	([Pro	perty[Source Modification]=='fals	e')			
	Format:	MBZ					
37:36	Source 0 Mc	difier					
	Exists If: ([Property[Source Modification]=='tru			ıe')			
	Format:	SrcM	od				
35	Reserved						
	Format:			MBZ			
34	Flag Register Number This field contains the flag register number for instructions with a non-zero Conditional Modifier.						
33	Flag Subregister Number This field contains the flag subregister number for instructions with a non-zero Conditional Modifier.						
32	Reserved						
	Format:			MBZ			
31:0	Header						
	11						

Linear Internelation Irm



and - Logic And									
Source:	EuIsa								
Length Bias:	4								
Description									
The and instr	uction perfor	ms compon	ent-wise lo	gic AND operation	between src0 and src1 and stores				
the results in dst.									
Register source operands can use source modifiers:									
Any source m	nodifier is nur	neric, optior	ally changi	ing a source value	s to -s. abs(s), or -abs(s) before the				
AND operation.									
Format:	Format:								
Source modi	fier is not allo	owed if sour	ce is an acc	cumulator.					
				Restriction					
Restriction: S	ource modifie	er is not allo	wed if sour	ce is an accumulate	or.				
				Suntay					
[(pred)] ar	d[gmod] (eved dize)	rea rea	reg [(pred)] an	d[amod] (even size) rea rea	imm 3.2			
[(pred)] al		exec_size)	IEG IEG	ieg [(pred)] an					
Pseudocode									
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src0.chan[n] & src1.chan[n]; } }</pre>									
Predication	Conditional	Modifier S	aturation	Source Modifier					
Y	Y	N N							
Src Types Dst Types									
*B,*W,*D *E	8,*W,*D								
DWord	Bit	Description							
03	127:64 ImmSource								
	Exists If: ([ImmSource][Src1.RegFile]=='IMM')				ile]=='IMM')				
	Format: EU_INSTRUCTION_SOURCES_REG_IMM				RCES_REG_IMM				
	127:64	RegSource							
		Exists If: ([RegSource][Src1.RegFile]!='IMM')							
		Format: EU_INSTRUCTION_SOURCES_REG_REG							
	63:32	63:32 Operand Controls							
		Format: EU_INSTRUCTION_OPERAND_CONTROLS							
	31:0 Header								
	Format: EU_INSTRUCTION_HEADER								


not - Logic Not								
Source:	EuIsa	a						
Length Bias:	4							
Description								
The not instruction of the results in dst.	The not instruction performs logical NOT operation (or one's complement) of src0 and storing the results in dst.							
This operation modifiers sho	on does not pould be used.	oroduce sig	n or overflov	w conditions. Only	the .e/.z or .ne/.nz conditional			
A register sou Any source r the NOT ope	urce operanc modifier is nu ration.	l can use a s imeric, optio	source modi onally chang	fier: ing a source value	s to -s, abs(s), or -abs(s) before			
Format: [(pred)] not[.cmod] (exec	_size) dst sr	c0					
				Restriction				
Restriction: S	ource modifi	er is not alle	owed if sour	ce is an accumulate	or.			
Suntax								
[(pred)] no	ot[.cmod] (exec_size) reg reg	[(pred)] not[.c	mod] (exec_size) reg imm32			
Decude es de								
Evaluate(Wr src0.chan[r	rEn); for (n]; } }	n = 0; n	< exec_si	ze; n++) { if	(WrEn.chan[n]) { dst.chan[n] = ~		
Predication	Conditiona	l Modifier	Saturation	Source Modifier				
Y	N		Y	Y				
Src Types D	st Types							
*B,*W,*D *E	8,*W,*D							
DWord	Bit			Dese	cription			
03	127:64	ImmSourc	e					
		Exists If:	([Opera	and Controls][Src0.	RegFile]=='IMM')			
		Format:	EU_INS	TRUCTION_SOURC	CES_IMM32			
	127:64	RegSource	9					
		Exists If: ([Operand Controls][Src0.RegFile]!='IMM')						
		Format:	EU_INS	STRUCTION_SOURC	LES_KEG			
	63:32	Operand (Format:	EU_INS	TRUCTION_OPERA	ND_CONTROLS			
	31:0	Header	I					
		Format:	E	U_INSTRUCTION_H	IEADER			



or - Logic Or								
Source:	EuIsa	1						
Length Bias:	4							
Description								
The or instruct results in dst. This operatic	The or instruction performs component-wise logic OR operation between src0 and src1 and stores the results in dst. This operation does not produce sign or overflow conditions. Only the .e/.z or .ne/.nz conditional							
Register sour Any source n the OR opera	Register source operands can use source modifiers: Any source modifier is numeric, optionally changing a source value s to -s, abs(s), or -abs(s) before the OR operation.							
[(pred)] or[.c	mod] (exec_s	ize) dst src0 sr	:1					
				Restriction				
Restriction: So	ource modifi	er is not allowe	d if sour	ce is an accumulate	or.			
	Syntax							
[(pred)] or	[.cmod] (e	xec_size) re	g reg r	reg [(pred)] or[.cmod] (exec_size) reg reg imm	n32		
Pseudocode								
Evaluate(Wr src0.chan[n	En); for (] src1.c	n = 0; n < han[n]; } }	exec_si	.ze; n++) { if	(WrEn.chan[n]) { dst.chan[n]] =		
Predication	Conditional	Modifier Sat	uration	Source Modifier				
Y	N	Y		Υ				
Src TypesD*B,*W,*D*E	st Types 3,*W,*D							
DWord	Bit			Des	cription			
03	127:64	ImmSource Exists If:	([Imn	nSource][Src1.RegFi	ile]=='IMM')			
	1 27.04	Pormat.						
	127:64	Frists If	([Reg	Source][Src1 ReaFi	e] ='IMM')			
		Format:	EU IN	NSTRUCTION SOUF	RCES_REG_REG			
	63:32	Operand Cor Format:	EU_IN	STRUCTION_OPERA	AND_CONTROLS			
	31:0	Header						
		Format:	E	U_INSTRUCTION_H	HEADER			



			xor	- Logic Xo	or			
Source:	EuIsa	1						
Length Bias:	4							
Description								
The xor instru the results in	The xor instruction performs component-wise logic XOR operation between src0 and src1 and stores the results in dst.							
This operation modifiers sho	on does not p ould be used.	broduce sign	or overflow	v conditions. Only	the .e/.z or .ne/.nz conditional			
Register sour Any source n the XOR oper	Register source operands can use source modifiers: Any source modifier is numeric, optionally changing a source value s to -s, abs(s), or -abs(s) before the XOR operation.							
Format: [(pred)] xor[.(cmod] (exec_	size) dst src	0 src1					
				Restriction				
Restriction: So	ource modifi	er is not allo	wed if sour	ce is an accumulate	or.			
				Syntax				
Syntax								
Pseudocode								
Evaluate(Wr src0.chan[n	En); for (] ^ srcl.c	n = 0; n han[n]; }	< exec_si }	ze; n++) { if	(WrEn.chan[n]) { dst.chan[n] =		
Predication	Conditional	Modifier S	Saturation	Source Modifier				
Y	N	١	(Y				
Src Types D	st Types							
*B,*W,*D *B	3,*W,*D							
DWord	Bit			Des	cription			
03	127:64	ImmSourc	e ([Imm	Sourcol[Src1 PogE	ilo] 'INANA')]		
		Format:	FU IN	ISTRUCTION SOUR	RCFS RFG IMM			
	127.64	RegSource						
	127.04	Exists If:	([Reg	Source][Src1.RegFi	le]!='IMM')			
		Format:	EU_IN		RCES_REG_REG			
	63:32	Operand C	ontrols					
		Format:	EU_INS	STRUCTION_OPERA	AND_CONTROLS			
	31:0	Header						
		Format:	E	U_INSTRUCTION_H	HEADER			



MEDIA_CURBE_LOAD								
Source:		Rend	erCS					
Length Bias: 2								
DWord	Bit				Descriptio	on		
0	31:29	Command	Command Type					
		Default Value: 3h GFXPIPE					XPIPE	
		Format:				ОрСо	de	
	28:27	Pipeline						
		Default Val	ue:			2	h Media	
		Format:				0	pCode	
	26:24	Media Com	nmand Opcode					1
		Default Val	ue:		0h MEDIA_CURE	BE_LOA	ND	
		Format: OpCode						
	23:16	SubOpcode						
		Default Value: 1h MEDIA_CURBE_LOAD SubOp						
		Format:		ОрС	ode			
	15:0	DWord Ler	ngth					
		Format:	=	=n To	tal Length - 2			
		Value		Na	ime		Description	
		2h	DWORD_COUNT_I	n [De	fault]		Excludes DWord (0,1)	
1	31:0	Reserved	•				-	
		Format:				I	MBZ	
2	31:17	Reserved						
		Format:				I	MBZ	
	16:0	CURBE Tot	al Data Length					
		Format:			U17 In Bytes			
								
		TI: (1)			Description			
		This field p	rovides the length i	in byt aliar	es of the CURBE	data. De Obi	ect Data Start Address As	
		the CURBE	data are sent direct	tly to	ROB, range is lim	nited to	CURBE Allocation Size.	
		This field n	nust be DWord (32-	byte)	aligned.			



			MEDIA_CURBE_LOAD						
3	31:0	CURBE Data Start Address							
	Format: DynamicStateOffset[31:0] CURBE								
	Description Specifies the 32-byte (DWord) aligned address of the CURBE data. This pointer is relative to the Dynamics Base Address.								
		Value Name							
				<u> </u>]					
			Programming Notes						
		ress based) Cache reusing the same							
		reused withi	in the same 64-byte cacheline.						



		MEDIA	_INT		RIPT	OR_LOAD			
Source:		RenderCS							
Length E	Bias:	2	2						
A Media storage	_State_ is clear	Flush should be ed.	used bef	fore this command to ensure	that the	e temporary Interface	e Descriptor		
DWord	Bit		Description						
0	31:29	Command Type	e						
		Default Value:			3h GFX	(PIPE			
		Format:			OpCoc	le			
	28:27	Pipeline							
		Default Value:			2h	Media			
		Format:			Op	Code			
	26:24	Media Comma	nd Opco	ode					
		Default Value:		0h MEDIA_INTERFACE_DES	CRIPTO	R_LOAD			
		Format:		OpCode					
	23:16	SubOpcode	SubOpcode						
		Default Value:	2	h MEDIA_INTERFACE_DESCR	IPTOR_L	OAD SubOp			
		Format:	С	DpCode					
	15:0	DWord Length							
		Format:		=n Total Length - 2					
		Value		Name		Descrip	tion		
		2h DW	ORD_CC	DUNT_n [Default]		Excludes DWord (0,	1)		
1	31:0	Reserved							
		Format:			N	1BZ			
2	31:17	Reserved					1		
		Format:			Ν	1BZ			
	16:0	Interface Descr	iptor To	otal Length					
		Format:		U17 In bytes					
		This field provid	es the le	ength in bytes of the Interface	e Descri	ptor data. This field n	nust have the		
		same alignment aligned. As the I Allocation Size	as the Ir nterface	nterface Descriptor Data Star Descriptor data are sent dire	ctly to	ss. It must be DQWoi ROB, range is limited	to CURBE		
		Value		Name	•				
		[32,1024]	[1,32	2] interface descriptor entries					



		MEDIA	_INTERFACE_DESCRIPTO	R_LOAD					
3	31:0	Interface Descriptor Data Start Address							
		Format: DynamicStateOffset[31:0]INTERFACE_DESCRIPTOR_DATA							
	DescriptionThis bit specifies the <u>32-byte</u> aligned address of the Interface Descriptor data. This pointer is relative to the Dynamics Base Address.								
	Value								
			Programming Notes						
		Driver must invalidate the vertex fetch cache thru the VF(address based) Cache Invalidation Enable thru a PIPE_CONTROL command prior to reusing the same graphics memory space.							
		VF cache inva reused within	alidation must be done when any graphics i the same 64-byte cacheline.	memory space is					



		MED	IA_		Т		
Source:		RenderCS					
Length E	Bias:	2					
DWord	Bit			Descriptio	on		
0	31:29	Command Type					
		Default Value: 3h GFXPIPE					
		Format:			OpCode		
	28:27	Media Command Pipeline					
		Default Value:			2h Me	edia	
		Format:			OpCo	de	
	26:24	24 Media Command Opcode					
		Default Value: 1h MEDIA_OBJECT					
		Format: OpCode					
	23:16 Media Command Sub-Opcode						
		Default Value: 0h MEDIA_OBJECT SubOp					
		Format:	Format: OpCode				
15:0 DWord Length							
		Default Value:	h DWORD_C	OUNT_n			
	Format: =n Total Length - 2						
		Generic Mode: DWord Length = N DW (equivalent to 63 8-DW registe When both inline and indirect data registers must be less than 112 (with both inline data length N and individually). The minimal inline data	ludes DWords 0,1 neric Mode: DWord Length = N+4, where N is in the range of [0,504]. The maximum ((equivalent to 63 8-DW registers). nen both inline and indirect data are fetched for this command, the total size in 8-DW isters must be less than 112 th both inline data length N and indirect data length rounded up to 8-DW aligned ividually). The minimal inline data length is 0.				
1	31:8	Reserved					
	7:6	Reserved					
		Format:			MBZ		
	5	Reserved					
		Format:			MBZ		
	4:0	Interface Descriptor Offset					
		Format:				U5	
		This field specifies the offset from the interface descriptor base pointer to the interface descriptor which will be applied to this object. It is specified in units of interface descriptors.					
2	31	Children Present					
		Format:		E	nable		
		Indicates that the root thread may	send	spawn messa	ages to spa	wn child threads	and/or



			M	DIA_OBJECT				
	synchroni If Childre receives a If Childre return UR signals UI thread. In order a	Synchronized root threads. If Children Present is not set, TS signals VFE to dereference the URB handle immediately after it receives acknowledgement from TD that the thread is dispatched. If Children Present is set, the URB handle is forwarded to the root thread and serves as the return URB handle for the root thread. TS does not signal deference at the time of dispatch. TS signals URB handle deference only when it receives a resource dereference message from the thread. In order avoid deadlock, such dereference must be issued once and only once for each URB handle.						
30:25	Reserved							
	Format:	Format: MBZ						
24	Thread S This field based on	Thread Synchronization This field when set indicates that the dispatch of the thread originated from this command is based on the "spawn root thread" message.						
	Value	No thread ar	- chroni-	Name				
	0	Thread dispat	ch is sv	nchronized by the 'snawn r	oot thread' message			
22	Pocorried	Inread dispatch is synchronized by the spawn root thread message						
25	Format: MBZ							
22	Reserved	Reserved						
	Format: MBZ							
21	Use Score This field Only whe cleared, the O	eboard specifies whet n this field is s he thread asso /alue	her the et, the s ociated v Not us	thread associated with this scoreboard control fields in with this command bypasse ing scoreboard	command uses hardware scoreboard. the VFE Dword are valid. If this field is s hardware scoreboard. Name			
	1		Using s	scoreboard				
20	Reserved							
	Format:				MBZ			
19	Reserved							
	Format:				MBZ			
18:17	Half-Slice This field	e Destination selects the ha	Select If slice t	hat this thread must be sen	t to.			
	Value	Name			Description			
	10b	Half-Slice 1		Cannot be used in product	ts without a Half-Slice 1.			
	01b	Half-Slice 0						
	00b	Either half-sli	се	Hardware will choose the s	lice based on load.			
				Programming Note	25			
	If "Either	half-slice" is s	elected	then the Slice Destination S	Select must also specify "Either slice".			
	30:25 24 23 22 21 20 19 18:17	Synchroni If Childre receives a If Childre return UR signals UI thread. In order a30:25Reserved Format:24Thread S This field based on Value24Thread S This field based on Value24Reserved Format:24Reserved Format:24Reserved Format:21Reserved Format:21Reserved Format:21Reserved Format:21Reserved Format:21Reserved Format:21Reserved Format:21Reserved Format:21Nas Score Format:21Nas Score Format:21Nas Score Format:21Nas Score Format:21Nas Score Format:22Reserved Format:30:25Nas Format:30:26Nas Format:30:27Nas Format:30:28Nas Format:30:29Reserved Format:30:20Reserved Format:30:30Nas Format:30:31Nas Format:30:32Nas Format:30:33Nas Format:30:34Nas Format:30:35Nas Format:30:35Nas Format:30:35Nas Format:30:35Nas Format:30:35Nas Format:30:35Nas Format:30:35Nas Format:30:35Nas Format:30:35 <td>synchronized root threat If Children Present is no receives acknowledgem If Children Present is sereturn URB handle for the signals URB handle defet thread. 30:25 Reserved Format: 7 24 Thread Synchronization This field when set indice based on the "spawn root value 0 0 No thread synchronization 1 Thread dispate 23 Reserved Format: 7 24 Reserved Format: 7 23 Reserved Format: 7 24 Reserved Format: 7 23 Reserved Format: 7 24 Reserved Format: 7 25 Reserved Format: 7 26 Reserved Format: 7 10 Reserved Format: 7 19 Reserved Format: 1 19 Reserved Format:<td>MIE Synchronized root threads. If Children Present is not set, T receives acknowledgement from If Children Present is set, the U Treceives acknowledgement from If Children Present is set, the U Teceives acknowledgement from If Children Present is set, the U Treceives acknowledgement from If Children Present is set, the U Teceives acknowledgement from If Children Present is set, the U Thread Synchronization This field when set indicates the based on the "spawn root thread Value 0 No thread synchronization This field specifies whether the Second Thread dispatch is sy 23 Reserved Format: Image: Seconeboard 21 Use Scoreboard This field specifies whether the Only when this field is set, the secleared, the thread associated of Value 20 Reserved Format: Image: Seconeboard 11 Using seconeboard 12 Not us 13 Image: Seconeboard This field selects the half slice to Image: Seconeboard 13 Thalf-Slice Destination Select 14 Value Name</td><td>MEDIA_OBJECT synchronized root threads. If Children Present is not set, TS signals VFE to dereference receives acknowledgement from TD that the thread is disp if Children Present is set, the URB handle is forwarded to return URB handle deference only when it receives a resorthread. In order avoid deadlock, such dereference must be issued or Thread. In order avoid deadlock, such dereference must be issued or thread. 30:25 Reserved Format: If Children Present is synchronization This field when set indicates that the dispatch of the thread based on the "spawn root thread" message. Name 0 No thread synchronization 1 Thread Synchronization 1 Thread dispatch is synchronized by the 'spawn r 23 Reserved Format: If Children Presend is set, the scoreboard control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared. 10 Not using scoreboard 11 Using scoreboard 12 Reserved Format: If Children Preserved 13 Using scoreboard 14 Using scoreboard</td></td>	synchronized root threat If Children Present is no receives acknowledgem If Children Present is sereturn URB handle for the signals URB handle defet thread. 30:25 Reserved Format: 7 24 Thread Synchronization This field when set indice based on the "spawn root value 0 0 No thread synchronization 1 Thread dispate 23 Reserved Format: 7 24 Reserved Format: 7 23 Reserved Format: 7 24 Reserved Format: 7 23 Reserved Format: 7 24 Reserved Format: 7 25 Reserved Format: 7 26 Reserved Format: 7 10 Reserved Format: 7 19 Reserved Format: 1 19 Reserved Format: <td>MIE Synchronized root threads. If Children Present is not set, T receives acknowledgement from If Children Present is set, the U Treceives acknowledgement from If Children Present is set, the U Teceives acknowledgement from If Children Present is set, the U Treceives acknowledgement from If Children Present is set, the U Teceives acknowledgement from If Children Present is set, the U Thread Synchronization This field when set indicates the based on the "spawn root thread Value 0 No thread synchronization This field specifies whether the Second Thread dispatch is sy 23 Reserved Format: Image: Seconeboard 21 Use Scoreboard This field specifies whether the Only when this field is set, the secleared, the thread associated of Value 20 Reserved Format: Image: Seconeboard 11 Using seconeboard 12 Not us 13 Image: Seconeboard This field selects the half slice to Image: Seconeboard 13 Thalf-Slice Destination Select 14 Value Name</td> <td>MEDIA_OBJECT synchronized root threads. If Children Present is not set, TS signals VFE to dereference receives acknowledgement from TD that the thread is disp if Children Present is set, the URB handle is forwarded to return URB handle deference only when it receives a resorthread. In order avoid deadlock, such dereference must be issued or Thread. In order avoid deadlock, such dereference must be issued or thread. 30:25 Reserved Format: If Children Present is synchronization This field when set indicates that the dispatch of the thread based on the "spawn root thread" message. Name 0 No thread synchronization 1 Thread Synchronization 1 Thread dispatch is synchronized by the 'spawn r 23 Reserved Format: If Children Presend is set, the scoreboard control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared. 10 Not using scoreboard 11 Using scoreboard 12 Reserved Format: If Children Preserved 13 Using scoreboard 14 Using scoreboard</td>	MIE Synchronized root threads. If Children Present is not set, T receives acknowledgement from If Children Present is set, the U Treceives acknowledgement from If Children Present is set, the U Teceives acknowledgement from If Children Present is set, the U Treceives acknowledgement from If Children Present is set, the U Teceives acknowledgement from If Children Present is set, the U Thread Synchronization This field when set indicates the based on the "spawn root thread Value 0 No thread synchronization This field specifies whether the Second Thread dispatch is sy 23 Reserved Format: Image: Seconeboard 21 Use Scoreboard This field specifies whether the Only when this field is set, the secleared, the thread associated of Value 20 Reserved Format: Image: Seconeboard 11 Using seconeboard 12 Not us 13 Image: Seconeboard This field selects the half slice to Image: Seconeboard 13 Thalf-Slice Destination Select 14 Value Name	MEDIA_OBJECT synchronized root threads. If Children Present is not set, TS signals VFE to dereference receives acknowledgement from TD that the thread is disp if Children Present is set, the URB handle is forwarded to return URB handle deference only when it receives a resorthread. In order avoid deadlock, such dereference must be issued or Thread. In order avoid deadlock, such dereference must be issued or thread. 30:25 Reserved Format: If Children Present is synchronization This field when set indicates that the dispatch of the thread based on the "spawn root thread" message. Name 0 No thread synchronization 1 Thread Synchronization 1 Thread dispatch is synchronized by the 'spawn r 23 Reserved Format: If Children Presend is set, the scoreboard control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared, the thread associated with this control fields in cleared. 10 Not using scoreboard 11 Using scoreboard 12 Reserved Format: If Children Preserved 13 Using scoreboard 14 Using scoreboard			



		MEDI	A_OBJECT						
	16:0	Indirect Data Length							
		Format:	U17 In bytes						
		This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Data Start Address field is ignored. This field must have the same alignment as the Indirect Object Data Start Address. It must be DQWord (32-byte) aligned. As the indirect data are sent directly to URB, range is limited to 496 DW. When both inline and indirect data are fetched for this command, the total size in 8-DW registers must be less than 112 (with both inline data length and indirect data length rounded up to 8-DW aligned).							
3	31:0	Indirect Data Start Address							
		Format: GraphicsA	ddress[31:0]						
			Description						
		This field specifies the Graphics Memory starting address of the data to be loaded into the kernel for processing. This pointer is relative to the Indirect Object Base Address .							
		Hardware ignores this field if indirect data is not present.							
		Alignment of this address depends of	on the mode of opera	ation.					
		This field specifies the DWord aligne	d address of the indi	rect data.					
		Value		Name					
		Pro	gramming Notes						
		Driver must invalidate the vertex fetch cache through the VF(address based) Cache Invalidation Enable through a PIPE_CONTROL command prior to reusing the same graphics memory space. VF cache invalidation must be done when any graphics memory space is reused within the same 64-byte cacheline.							
		Bits 31:29 MBZ							
4	31:25	Reserved							
		Format:		MBZ					
	24:16	Scoredboard Y							
		Format:		U9					
		This field provides the Y term of the s	coreboard value of th	he current thread.					
	15:9	Reserved							
		Format:		MBZ					
	8:0	Scoreboard X							



		MEDIA_OB.	JECT					
		Format:			U9			
		This field provides the X term of the scoreboard	This field provides the X term of the scoreboard value of the current thread.					
5	31:20	Reserved						
		Format:		MBZ				
	19:16	6 Scoreboard Color						
	Format: U4							
		This field specifies which dependency color the dependency scoreboard control.	field specifies which dependency color the current thread belongs to. It affects the endency scoreboard control.					
	15:8	Reserved						
		Format:		MBZ				
	7:0	Scoreboard Mask						
		Format:	Boolean					
Each bit indicates the corresponding dependency scoreboard is dependent on. The AND'd with the corresponding Scoreboard Mask field in the MEDIA_VFE_STATE of the formula of the corresponding scoreboard mask field in the MEDIA_VFE_STATE of the second score sco					ependent on. This field is A_VFE_STATE command.			
				t o maps				
6n	31:0	Inline Data Generic Mode: The format of this data is specific data; it merely passes it to the kernel for proces data must not exceed 112 registers.	ed by softv sing. The to	vare. Hai otal size	rdware does not interpret this for the inline data and indirect			

Doc Ref # IHD-OS-VLV-Vol2 pt2-04.14



MEDIA_OBJECT_PRT

Source: RenderCS

Length Bias:

This command is for generating a Persistent Root Thread for the media pipeline. It only supports loading of inline data but not indirect data. The command can be used in all VFE modes, except VLD mode.

For simplification, _PRT command has a fixed size of 16 DWORD

DWord	Bit				Descripti	on			
0	31:29	Command	Туре						
		Default Val	ue:			3h C	h GFXPIPE		
		Format:				ОрС	Code		
	28:27	Pipeline							
		Default Value:					2h Me	dia	
		Format:					ОрСос	de	
	26:24	Media Com	mand Opcode						
		Default Val	ue:		1h MEDIA_OB	JECT_	PRT		
		Format:			OpCode				
	23:16	SubOpcode	SubOpcode						
		Default Val	ue:	2h M	EDIA_OBJECT_I	PRT S	ubOp		
		Format:		ОрСо	ode				
	15:0	DWord Len	DWord Length						
		Format: =n Total Length - 2							
		Note: Regain must fit with	rdless of the mode, hin 16 dwords.	inline	data must be p	oreser	nt in thi	is command. The command size	
		Value		Nar	ne			Description	
		0Eh	DWORD_COUNT_n	[Defa	ault]		Exc	cludes DWord (0,1)	
1	31:6	Reserved							
		Format:					MBZ		
	5	Reserved							
		Format:					MBZ		
	4:0	Interface Descriptor Offset							
		Format:						U5	
		This field sp	ecifies the offset fro	m the	e interface desc	riptor	base p	pointer to the interface descriptor	
		which will be applied to this object. It is specified in units of interface descriptors.							
2	31	Children Pr	resent						
		Format:				Enabl	е		
		Indicates the	at the root thread m	nay se	nd spawn mess	sages	to spav	wn child threads and/or	
		synchronize	a root threads. Present is not set It	S sian	als VFF to dere	feren	e the I	JRB handle immediately after it	
		receives ack	nowledgement from	n TD t	hat the thread	is disp	patched	d.	



			MEDIA_OBJEC	T_PR	Т			
		If Children Pi return URB ha signals URB h thread. In order avoi handle.	resent is set, the URB handle is forwa andle for the root thread. TS does no nandle deference only when it receiv id deadlock, such de-reference must	arded to t ot signal d es a resou be issued	the root thread and serves as the deference at the time of dispatch. TS urce dereference message from the d once and only once for each URB			
	30:24	Reserved						
		Format:			MBZ			
	23	PRT_Fence N	leeded					
		Format:		Enable	2			
		MEDIA_OBJECT_PRT. The PRT_Fence prevents additional threads following this persistent root thread until a thread spawn message is sent. The PRT_Fence is generated on first dispatch of the persistent root, as well as on re-dispatches of the persistent root after context restore.						
	22	PRT_FenceTy This field spetthe end of the these root the to 1, the fenc command. No (by the PRT). This field is c	ype cifies the type of fence the PRT threa e root thread queue. It will block the reads to be populated through VFE e is set at the entry of VFE, similar to o more command can go into the m only valid when PRT_Fence Needed i	ad uses. If dispatch to the roc the fenc edia pipe s set to 1.	f this field is set to 0, the fence is set at of the next root thread, but allowed of thread queue in TS. If this field is set the set by the MEDIA_STATE_FLUSH e until a thread spawn message is sent . Otherwise, it is ignored by hardware.			
		Value	Name		Description			
		0h	Root thread queue	Root th	read queue fence			
		1h	VFE state flush	VFE stat	te flush fence			
	21:0	Reserved						
		Format:			MBZ			
3	31:0	Reserved						
		Format:			MBZ			
415	31:0	Inline Data						
Format: U32								



		MEDI	A_C)B	JECT_W		KER		
Source:		RenderCS							
Length E	Bias:	2							
DWord	Bit				Descriptio	n			
0	31:29	Command Type			-				
		Default Value:				3h G	GFXPIPE		
	Format: OpCode						Iode		
	28:27	Pipeline							
		Default Value:				2h Media			
		Format:					OpCode		
	26:24	Media Command Opcode							
		Default Value: 1h			MEDIA_OBJECT_	WAL	LKER		
		Format: Or			Code				
	23:16	SubOpcode							
		Default Value: 03h ME			DIA_OBJECT_WAL	_KER	≀ SubOp		
		Format: OpCode			de				
	15:0	DWord Length							
		Default Value:			0Fh DWORD_C	NUC	IT_n		
		Format:			=n Total Length	า - 2			
		common for all threads generated by greater data is not present. It should optional for this command.	erated d be n	d fro	d that unlike oth	d, If t er m	this field is 15, it indicates that inline nedia object command, inline data is		
1	31:8	Reserved							
	7:6	Reserved							
		Format: Reserved							
	5	Reserved							
		Format:					MBZ		
	4:0	Interface Descriptor Offset	t						
		Format:	Format: U5						
		This field specifies the offset which will be applied to this	t from objec	n the ct. It	interface descri is specified in u	ptor nits d	base pointer to the interface descriptor of interface descriptors.		
2 31 Children Present									
		Format:			Boole	an			
		Indicates that the root thread synchronized root threads. I handle immediately after it i Children Present is set, the U URB handle for the root thre	d may f Child receiv JRB h ead. T	y sei dren 'es a andl 'S dc	nd spawn messa Present is not s cknowledgemen le is forwarded to bes not signal de	ges t et, T nt fro o the ferei	to spawn child threads and/or 'S signals VFE to dereference the URB om TD that the thread is dispatched. If e root thread and serves as the return nce at the time of dispatch. TS signals		



			ME	DIA_OBJECT_WAL	KER				
	URB handle deference only when it receives a resource dereference message fro order avoid deadlock, such dereference must be issued once and only once for each								
	30:25	Reserved							
		Format:			MBZ				
24 Thread Synchronization This field when set indicates that the dispatch of the thread originated from this common based on the "spawn root thread" message. Value Name									
		0	No thread syr	nchronization					
		1	Thread dispat	tch is synchronized by the 'spawn i	oot thread' message				
	23:22	Reserved	1						
		Format:			MBZ				
	21 Use Scoreboard This field specifies whether the thread associated with this command uses hardware score Only when this field is set, the scoreboard control fields in the VFE Dword are valid. If this cleared, the thread associated with this command bypasses hardware scoreboard.								
	0 Not using scoreboard								
		1		Using scoreboard					
	20:17	Reserved	1						
		Format:			MBZ				
	16:0	Indirect	Data Length						
		Format:		U17 in bytes					
	This field provides the length in bytes of the indirect data. A value zero indicates that ind data fetching is disabled - subsequently, the Indirect Data Start Address field is ignored. This field must have the same alignment as the Indirect Object Data Start Address. It must be DQWord (32-byte) aligned. As the indirect data are sent directly to URB, rang limited to 496 DW. When both inline and indirect data are fetched for this command, the size in 8-DW registers must be less than or equal to 63 (with both inline data length and data length rounded up to 8-DW aligned).								
3	31:0	Indirect	Data Start Ad	dress					
		Format: GraphicsAddress[31:0]							
		the kern Hardwa Alignme	el for processi re ignores this ent of this addi	Graphics Memory starting address ng. This pointer is relative to the Ir field if indirect data is not present ress depends on the mode of oper	of the data to be loaded into direct Object Base Address. ation.				
		It is the l	DWord aligned	address of the indirect data.					

Г



		MEDIA_C	DBJECT_\	WALKER					
		Value	Name	Description					
		[0 - 512MB]		(Bits 31:29 MBZ)					
4	31:0	Reserved							
		Format:		MBZ					
5	7:0	Scoreboard Mask							
		Format:	В	oolean					
		Each bit indicates the corresponding Sc AND'd with the corresponding Sc generated by this walker comman Bit n (for n = 07): Scoreboard in	tes the corresponding dependency scoreboard is dependent on. This field is e corresponding Scoreboard Mask field in the MEDIA_VFE_STATE. All threads this walker command share the same dynamic mask. 07): Scoreboard n is dependent, where bit 0 maps to n = 0.						
6	31	Dual Mode							
		Format:	В	oolean					
	30	Repel							
		Format:	oolean						
		Due une suite a blacker							
		Programming Notes							
		Repei should not be combined w							
	29	Reserved		N/D7					
		Hormat: MBZ							
	28	Reserved		MPZ					
	27.24	Color Court Minus One		MBZ					
	27.24	Format:		114					
		This field specifies the number of repeat of the inner most loop of the walker. Each repeated walk position is assigned with an incremental Color number. The Color number together with the X and Y position of the thread is used for dependency scoreboard control. Usage Example: This allows multiple sets of dependency threads to be dispatched.							
	23:21	Reserved							
		Format:		MBZ					
	20:16	Middle Loop Extra Steps							
		Format:		U5					
	15:14	Reserved							
		Format:		MBZ					
	13:12	Local Mid-Loop Unit Y							
		Format:		S1					
	11:10	Reserved							
		Format:		MBZ					



		MEDIA_OBJECT_WAI	.KER						
	9:8	Mid-Loop Unit X							
		Format:		S1					
	7:0	Reserved							
		Format:	MBZ						
7	31:26	Reserved							
		Format:	MBZ						
	25:16	Global Loop Exec Count							
		Format:	U10						
	15:10	Reserved							
		Format:	MBZ						
	9:0	Local Loop Exec Count							
		Format:	U10						
8	31:25	Reserved							
		Format:	MBZ						
	24:16	Block Resolution Y		·					
		Format:		U9					
		Vertical resolution of the local loop.							
	15.0	9 Reserved							
	13.5	Format:	MBZ						
	8.0	Block Resolution X							
	0.0	Format:		U9					
		Horizontal resolution of the local loop.							
9	31:25	Reserved							
		Format:	MBZ						
	24:16	Local Start Y		1					
		Format:		09					
		Starting vertical position of the local loop.							
	15:9	Reserved							
Format: MBZ									
	8:0	Local Start X	<u>.</u>						
		Format:		U9					
		Starting horizontal position of the local loop.							
10	10 31:25 Reserved								
		Format:	MBZ						
	24:16	Local End Y							



		MEDIA_OBJECT_WAL	KER						
		Format:	U9						
		Ending vertical position of the local loop.							
	15:9	Reserved							
		Format: MBZ							
	8:0	Local End X	i						
		Format:	U9						
		Ending horizontal position of the local loop.							
11	31:26	Reserved							
		Format:	MBZ						
	25:16	Local Outer Loop Stride Y							
		Format:	S9						
		Vertical stride of the local outer loop, in 2's complement.							
	15:10	Reserved							
		Format:	MBZ						
	9:0	Local Outer Loop Stride X							
		Format:	S9						
		Horizontal stride of the local outer loop, in 2's compleme	nt.						
12	31:26	Reserved							
		Format:	MBZ						
	25:16	Local Inner Loop Unit Y							
		Format:	S9						
		Vertical stride of the local inner loop, in 2's complement.							
	15:10	Reserved							
		Format:	MBZ						
	9:0	Local Inner Loop Unit X							
		Format:	S9						
		Horizontal stride of the local inner loop, in 2's compleme	nt.						
13	31:25	Reserved							
		Format:	MBZ						
	24:16	Global Resolution Y							
		Format:	U9						
		Vertical resolution of the global loop.							
	15:9	Reserved							



		MEDIA_OBJECT_WALI	KER		
		Format:	MBZ		
	8:0	Global Resolution X	•		
		Format:	L	19	
		Horizontal resolution of the global loop.			
14	31:26	Reserved			
			MBZ		
	25:16	Global Start Y		CO	
		Format:		59	
		Starting vertical location of the global loop, in 2 s compler	nent.		
	15:10	Reserved			
		Format:	MBZ		
	9:0	Global Start X			
		Format:		S9	
		Starting horizontal location of the global loop, in 2's comp	lement.		
15	31:26	Reserved			
		Format:	MBZ		
	25:16	Global Outer Loop Stride Y			
		Format:		S9	
		Vertical stride of the global outer loop, in 2's complement.			
	15:10	Reserved			
		Format:	MBZ		
	9:0	Global Outer Loop Stride X			
		Format:		S9	
		Horizontal stride of the global outer loop, in 2's compleme	ent.		
16	31:26	Reserved			
Format: MBZ 25:16 Global Inner Loop Unit Y					
		Vertical stride of the global inner loop, in 2's complement.			
	15:10	Reserved			
		Format:	MBZ		



MEDIA_OBJECT_WALKER							
	9:0	Global Inner Loop Unit X					
		Format:	S9				
		Horizontal stride of the global inner loop, in 2's complement.					
17n	31:0	Inline Data					



MEDIA_STATE_FLUSH

20	urc	:e:	

Length Bias:

This command updates the Message Gateway state. In particular, it updates the state for a selected Interface Descriptor.

This command can be considered same as a MI_Flush except that only media parser will get flushed instead of the entire 3D/media render pipeline. The command should be programmed prior to new Media state, curbe and/or interface descriptor commands when switching to a new context or programming new state for the same context.

With this command, pipelined state change is allowed for the media pipe.

It should be cautious when using this command when child_present flag in the media state is enabled. This is because that CURBE state as well as Interface Descriptor state are shared between root threads and child threads. Changing these states while child threads are generated on the fly may cause unexpected behavior. Combining with MI_ARB_ON/OFF command, it is possible to support interruptability with the following command sequence where interrupt may be allowed only when MI_ARB_ON_OFF is ON:

MEDIA_STATE_FLUSH

VFE_STATE // VFE will hold CS if watermark isn't met

RenderCS

2

MI_ARB_OFF // There must be at least one VFE command before this one

MEDIA_OBJECT MI_ARB_ON

DWord	Bit	Description							
0	31:29	Command	Туре						
		Default Val	ue:			3h GFXPIPE			
		Format:				ОрС	Code	2	
	28:27	Pipeline							
		Default Val	ue:				2h I	Vedia	
		Format:					ОрС	Code	
	26:24	Media Com	mand Opcode						
		Default Val	ue:		0h MEDIA_STA	TE_FLU	JSH		
		Format:			OpCode				
	23:16	SubOpcode	e A						
		Default Val	ue:	4h M	IEDIA_STATE_FL	USH S	JSH SubOp		
		Format:		OpCo	ode				
	15:0	DWord Ler	gth						
		Format:		=n To	tal Length - 2				
		Value		Na	me			Description	
		0h	DWORD_COUNT_	WORD_COUNT_n [Default]				Excludes DWord (0,1)	
1	31:9	Reserved	served						
		Format:	MBZ						
	8	Disable Pre	e-emption						
		Format:				Enabl	е		
		This bit cau	ses the video front	-end t	o ignore pre-en	nption	req	uests if set. If this bit is set then	



	MEDIA_STATE_FLUSH
	ARB_CHECK commands should not be used with it. A subsequent MEDIA_STATE_FLUSH command with this bit cleared will honor previous pre- emption requests.
	Reserved
	Format: MBZ
	 Watermark Required This is a single bit specifying if the MEDIA_STATE_FLUSH should stall further commands until there is enough room in a half-slice for the following thread group. The characteristics of the thread group are specified in the Interface Descriptor Offset. If set, the MEDIA_STATE_FLUSH stalls CS until there are enough threads in a half-slice, and enough SLM available in the same half-slice, and a free barrier if one is required. An Interface Descriptors can be updated after a Watermarked MEDIA_STATE_FLUSH only if it has not been used in the current context. Reusing an interface desciptor requires that this bit is clear to ensure the ID cache is reloaded. If clear, the MEDIA_STATE_FLUSH stalls CS until the TDL has dispatched the last thread, allowing the CURBE and Interface Descriptors to be updated by following commands.
!	0 Interface Descriptor Offset
	Format: U6
	This field specifies the offset from the interface descriptor base pointer to the interface descriptor which describes what resources are required to meet the watermark.



	MEDIA_VFE_STATE									
Source: RenderCS										
Length E	Bias:	2								
DWord	Bit					Descriptio	on			
0	31:29	Command	Туре				I			
		Default Val	ue:				3h GFX	(PIPE		
		Format:					OpCoc	le		
	28:27	Pipeline								
		Default Val	ue:				2h	Media		
		Format:					Op	oCode		
	26:24	Media Com	mand Opco	de						
		Default Val	ue:		0h MEDIA_VFE_STAT			E		
		Format:			OpCode					
	23:16	SubOpcode	e A		k					
		Default Val	ue:		0h M	/EDIA_VFE_STATE SubOp				
		Format:			ОрСс	ode				
	15:0	DWord Len	igth							
		Format:		=r	n Total	Length - 2				
		Value			Nam	۵		Description		
		06h	DWORD CC)UNT n	[Defai	ult]		Excludes DWord (0.1)		
1	31.10	Scratch Spa	ace Base Poi	ntor	[
Т	51.10	Format [.]	Cratch Space base Folliter							
		Specifies the 1k-byte aligned address offset to scratch space for use by the kernel. This pointer								
		relative to th	he General S	tate Ba	se Ado	dress.	I	, ,		
	9:4	Reserved								
		Format:					Ν	1BZ		



	3:0	Per Three	ad Scratch	Space						
		Format:				U4				
		Specifies the amount of scratch space allowed to be used by each thread. The driver								
		must all	ocate end	ugh conti	guous scratch spa	ace, pointed to by the Scratch Space				
		Pointer, to ensure that the maximum threads in the device each get Per Thread Scratch								
		Valu	е	Name		Description				
		[0,11]	[0,11] indicating [1k bytes, 12k bytes]							
			· ··· · ·		Programmi	ng Notes				
		thread s	scratch spa	ace is spe	s different from tr cified in powers o	f 2.				
2	31:16	Maximu	n Number	of Thread	ls					
		Format:		U16-1 rep	presenting thread co	punt				
		Range: [0, n-1] where n = (# EUs) * (# threads/EU). See <i>Graphics Processing Engine</i> for listing of #EUs and #threads in each device.								
		Specifies the maximum number of simultaneous root threads allowed to be active. Used to								
		avoid potential deadlock.								
		If child threads are not planning on being used then this field can be set to its maximum value and there will be no thread limit beyond what is currently available in the system: the maximum								
		value can include threads in slices that have been shut down for power reasons.								
		Programming Notes								
		MSB will be zero due to the range limit below.								
	15:8	Number of URB Entries								
		Format: U8								
		Specifies	the numbe	r of URB e	ntries that are used	by the unit.				
			Value		Name	Description				
		[0,64]				[0,64] Entries				
	7	Reset Ga	teway Tim	er						
			controls th	e reset of	the timestamp cour	Name				
		Oh	Maintain	ing the evi	sting timestamp sta					
		1h	Resetting	n relative ti	mer and latching th	ne global timestamp				
	6	Bypass G	atoway Co							
	0	This field	configures	Gateway t	o use a simple mes	sage protocol.				
		Value	5	,	•	Name				
		0h	Maintaining	g OpenGat	eway/ForwardMsg/	CloseGateway protocol (legacy mode)				



				MED	IA_VFE_STATE		
		1h Byp	assing Ope	nGatewa	ay/CloseGateway protocol		
	5	Reserved					
	4:3	Gateway M	MIO Access	s Contro	bl		
		The Gateway	allows me	nessages from EUs to read and write MMIO registers. This field limits this			
		feature for se	ecurity reas	ons	Nama		
		value	No M		Name		
		1	Rocor	word			
		2) read/w	wite to any address		
	2			J Teau/w			
	2	This bit indicates whether the VFE is in GPGPU mode (will expect GPGPU_OBJECT and GPGPU_WALKER commands) or MEDIA mode (will expect MEDIA_OBJECT and MEDIA_WALK commands)					
			Value		Name		
		0h			MEDIA Mode		
		1h			GPGPU Mode		
	1:0	Reserved					
3	31:8	Reserved					
	7:0	Reserved					
		Format:			MBZ		
4	31:16	URB Entry A	Ilocation S	Size			
		Format:			U16-1		
		Specifies the address for (URB Entry of than or equ Functions U If SLM is en	e length of URB starts Allocation S al to the nu nified Retur abled then	each UR after CU iize * Nu imber of rn Buffer the num	RB entry used by the unit, in 256-bit register increments - 1. ROB RBE Allocated region. Imber of URB Entries) + CURBE Allocation Size + 32) must be less f entries in the URB as described in vol5c.5 Shared r, under the section "URB Size". There of available entries will be 1/3 the maximum URB entries.		
		Value	Namo		Description		
			Name	LIRR En	try Alloc Size when SLM is disabled		
		[0,3040]			try Alloc Size when SLM is enabled		
		[0,552]			Alloc Size when Selvins chabled.		
					Programming Notes		
		When Inline allocation si If Indirect d the total All If both Inlin total space	e data is use ze must ma ata is being ocation Size e and Indire must be en	ed with N atch the used wi e * Num ect are b ough for	MEDIA_OBJECT or MEDIA_OBJECT_WALKER, then the URB entry Inline data size. ith MEDIA_OBJECT then the allocation size does not matter, but ber of URB Entries should be sufficient for the Indirect data. being used, then the allocation size must match the Inline and the r both the Indirect and Inline.		



			N	MEDIA_VFE_STATE					
	15:0	CURBE Alloca	tion Size						
		Format:		U12-1					
		Specifies the total length allocated for CURBE, in 256-bit register increments - 1. ROB address for CURBE starts at address 32. (URB Entry Allocation Size * Number of URB Entries) + CURBE Allocation Size + 32) must be less than or equal to the number of entries in the URB as described in vol5c.5 Shared Functions Unified Return Buffer, under the section "URB Size".							
		If SLM is enabled then the number of available entries will be 1/3 the maximum URB entries.							
		Value Name		Description					
		[0,2016]		CURBE Alloc Size when SLM is disabled					
		[0,992]		CURBE Alloc Size when SLM is enabled					
5	31	Scoreboard E	nable						
		This field enables and disables the hardware scoreboard in the Media Pipeline. If this field is cleared, hardware ignores the following scoreboard state fields.							
		Valu	e	Name					
		0h		Scoreboard disabled					
		1h		Scoreboard enabled					
	30	Scoreboard Type This field selects the type of scoreboard in use.							
		Value		Name					
		0h	Stalling	g scoreboard					
		1h	Non-st	talling scoreboard					
	29:8	Reserved	eserved						
		Format:		MBZ					
	7:0	Scoreboard N	lask						
		Format:		Enable[8]					
		Each bit indica based on the r Score n is enal	ites the con relative (X, pled.	rresponding dependency scoreboard is enabled. The scoreboard is Y) distance from the current threads' (X, Y) position. Bit n (for n = 07):					
6	31:28	Scoreboard 3	Delta Y						
		Format:		S3					
		Relative vertical distance of the dependent instance assigned to scoreboard 3, in the form of 2's compliment.							
	27:24	Scoreboard 3	Delta X						
		Format:		S3					
		Relative horizo 2's complimen	ontal distar it.	nce of the dependent instance assigned to scoreboard 3, in the form of					



1 . I. / VI	Scoreboard 2 Delta V							
	Format:	53						
	Relative vertical distance of the dependent instance of the dependent instance of the dependent instance compliment.	ce assigned to scoreboard 2, in the form o						
19:16	Scoreboard 2 Delta X							
	Format:	\$3						
	Relative horizontal distance of the dependent instance assigned to scoreboard 2, in the form of 2's compliment.							
15:12	Scoreboard 1 Delta Y							
	Format:	\$3						
	Relative vertical distance of the dependent instance of the dependent instance of the dependent instance compliment.	ce assigned to scoreboard 1, in the form o						
11:8	Scoreboard 1 Delta X							
	Format:	S3						
	Relative horizontal distance of the dependent instance assigned to scoreboard 1, in the form 2's compliment.							
7:4	Scoreboard 0 Delta Y							
	Format:	S3						
	Format: Relative vertical distance of the dependent instance compliment.	S3 ce assigned to scoreboard 0, in the form o						
3:0	Format: Relative vertical distance of the dependent instance compliment. Scoreboard 0 Delta X	S3 ce assigned to scoreboard 0, in the form o						
3:0	Format: Relative vertical distance of the dependent instance compliment. Scoreboard 0 Delta X Format:	S3 ce assigned to scoreboard 0, in the form of S3						
3:0	Format: Relative vertical distance of the dependent instance compliment. Scoreboard 0 Delta X Format: Relative horizontal distance of the dependent inst 2's compliment.	S3 ce assigned to scoreboard 0, in the form of S3 ance assigned to scoreboard 0, in the for						
3:0	Format: Relative vertical distance of the dependent instance compliment. Scoreboard 0 Delta X Format: Relative horizontal distance of the dependent inst 2's compliment. Scoreboard 7 Delta Y	S3 ce assigned to scoreboard 0, in the form of S3 ance assigned to scoreboard 0, in the for						
3:0 31:28	Format: Relative vertical distance of the dependent instance compliment. Scoreboard 0 Delta X Format: Relative horizontal distance of the dependent instance of the dependent instance 2's compliment. Scoreboard 7 Delta Y Format:	S3 ce assigned to scoreboard 0, in the form of S3 ance assigned to scoreboard 0, in the for S3						
3:0	Format: Relative vertical distance of the dependent instance compliment. Scoreboard 0 Delta X Format: Relative horizontal distance of the dependent instance of the	S3 ce assigned to scoreboard 0, in the form of S3 ance assigned to scoreboard 0, in the for S3 ce assigned to scoreboard 7, in the form of						
3:0 31:28 27:24	Format: Relative vertical distance of the dependent instance compliment. Scoreboard 0 Delta X Format: Relative horizontal distance of the dependent instance of the	S3 ce assigned to scoreboard 0, in the form of S3 ance assigned to scoreboard 0, in the for S3 ce assigned to scoreboard 7, in the form of						
3:0 31:28 27:24	Format: Relative vertical distance of the dependent instance compliment. Scoreboard 0 Delta X Format: Relative horizontal distance of the dependent instance compliment. Scoreboard 7 Delta Y Format: Relative vertical distance of the dependent instance compliment. Scoreboard 7 Delta X Format:	S3 ce assigned to scoreboard 0, in the form of S3 ance assigned to scoreboard 0, in the for S3 ce assigned to scoreboard 7, in the form of S3 ce assigned to scoreboard 7, in the form of S3						



	MEDIA_VFE_S	STATE				
	Format:	\$3				
	Relative vertical distance of the dependent insta compliment.	nce assigned to scoreboard 6, in the form of 2's				
19:16	Scoreboard 6 Delta X					
	Format:	S3				
	Relative horizontal distance of the dependent instance assigned to scoreboard 6, in the form of 2's compliment.					
15:12	Scoreboard 5 Delta Y					
	Format:	S3				
	Relative vertical distance of the dependent insta compliment.	nce assigned to scoreboard 5, in the form of 2's				
11:8	Scoreboard 5 Delta X					
	Format:	S3				
	Relative horizontal distance of the dependent instance assigned to scoreboard 5, in the form of 2's compliment.					
7:4	Scoreboard 4 Delta Y					
	Format:	\$3				
	Relative vertical distance of the dependent insta compliment.	nce assigned to scoreboard 4, in the form of 2's				
3:0	Scoreboard 4 Delta X					
	Format:	\$3				
	Relative horizontal distance of the dependent in 2's compliment.	stance assigned to scoreboard 4, in the form of				



MFC_AVC_PAK_OBJECT

Source: VideoCS

2

Length Bias:

The MFC_AVC_PAK_OBJECT command is the second primitive command for the AVC Encoding Pipeline. The same command is used for both CABAC and CAVLC modes. The MV Data portion of the bitstream is loaded as indirect data object.Before issuing a MFC_AVC_PAK_OBJECT command, all AVC MFX states need to be valid. Therefore the commands used to set these states need to have been issued prior to the issue of this command. MB record must be consecutive with no gaps, hence we do not need MB(x,y) in each MB command. Internal counter will keep track of the current MB address, starting from the Start_MB_In_Slice loaded at the beginning of each slice. MFC_AVC_PAK_OBJECT command follows the MbType definition like MFD. Many fields in this command are identical to that in VME output. This is intended to reduce software converting overhead from VME to PAK. Encoding statistical data such as the total size of the output bitstream are provided through MMIO registers. Software may access these registers through MI_STORE_REGISTER_MEM command.

Dvvora	Bit			Descript	tion		
0	31:29	Command Typ	e				
		Default Value:		3h PARALLEL_V	IDEO_PI	PE	
		Format:		OpCode			
	28:27	Pipeline					
		Default Value:	2	2h MFC_AVC_PA	AK_OBJE	СТ	
		Format:	(OpCode			
	26:24	Media Comma	nd Opcode				
		Default Value:			1h AVC	_ENC	
		Format:			OpCode	2	
	23:21	SubOpcode A					
		Default Value:				2h	
		Format:				OpCode	
	20:16	SubOpcode B				1	
		Default Value:				9h	
		Format:				OpCode	
	15:12	Reserved					
		Format:			1	MBZ	
	11:0	DWord Length					
		Format:		=n Length -2			
		Value		Name			
		0009h	DWORD COUNT n	[Default]			
1	31.10	Reserved		<u> </u>			
_	0 0	Format:			1	MBZ	
	9:0	Indirect PAK-N	IV Data Length		L		
		This field provid current MB (in a	les the length in byte any partitioning and	es of the indired subpartitioning	ct data, w form). A	which contains all the	e MVs for the s that indirect



		MFC_AVC_PAK_OBJE	СТ
		data fetching is disabled - subsequently, the Indirect PAK- This field must have the same alignment as the Indirect PA must be DW aligned (since each MV is 4 bytes in size). Driv MVsize (MVquantity in DXVA, exact size) *4 bytes per MV.	MV Data Start Address field is ignored. K-MV Data Start Address. This field ver has to derived this field from
2	31:29	Reserved	
		Format:	MBZ
	28:0	Indirect PAK-MV Data Start Address Offset This field specifies the memory starting address (offset) of Subsystem for processing. This pointer is relative to the M Address. Hardware ignores this field if indirect data is not Length is set to 0. It is a Dword aligned address in all AVC is 4 bytes in size.	the MV data to be fetched into PAK FC Indirect PAK-MV Object Base present, i.e. the Indirect PAK-MV Data encoding configuration, since each MV
		Value	Name
		[0,512MB)	
310	31:0	Inline Data All the required MB level controls and parameters for enco MFC_AVC_PAK_OBJECT command. It has a fixed size of 8 D next section.	ding are captured as inline data of the DWs. Its definition is described in the



MFC_MPEG2_PAK_OBJECT

Source: VideoCS

2

Length Bias:

The MFC_MPEG2_PAK_OBJECT command is the second primitive command for the MPEG-2 Encoding Pipeline. Different from AVC, the MV Data portion of the bitstream is loaded as part of MB control data.

Before issuing a MFC_MPEG2_PAK_OBJECT command, all MPEG2_MFX states need to be valid. Therefore the commands used to set these states need to have been issued prior to the issue of this command.

MB record must be consecutive with no gaps, hence we do not need MB(x,y) in each MB command. Internal counter will keep track of the current MB address, starting from the Start_MB_In_Slice loaded at the beginning of each slice.

MFC_ MPEG2_PAK_OBJECT command follows the MbType definition like MFD. Many fields in this command are identical to that in VME output. This is intended to reduce software converting overhead from VME to PAK.

Encoding statistical data such as the total size of the output bitstream are provided through MMIO registers. Software may access these registers through MI_STORE_REGISTER_MEM command.

DWord	Bit			Description				
0	31:29	Command Type						
		Default Value:		3h PARALLEL_VIDE	O_PI	PE		
		Format:		OpCode				
	28:27	Pipeline						
		Default Value:	2h M	FC_AVC_PAK_INSER	T_OE	BJECT		
		Format:	ОрСс	ode				
	26:24	Media Command Opcode						
		Default Value:			3h N	MPEG2		
		Format:			ОрО	Code		
	23:21	SubOpcode A						
		Default Value:				2h ENC		
		Format:				OpCode		
	20:16	SubOpcode B						
		Default Value:			9h M	EDIA_		
		Format:			ОрСо	ode		
	15:12	Reserved						
		Format:				MBZ		
	11:0	DWord Length						
		Default Value:	0	007h Excludes DWo	ord (0),1)		
		Format:	=	n Total Length - 2				
18	31:0	Inline Data All the required MB level cont	rols a	and parameters for e	encod	ling are captured as inline data of		



MFC_MPEG2_PAK_OBJECT	
the MFC_MPEG2_PAK_OBJECT command. It has a fixed size of 8 DWs. Its definition is described in the next section	



		MFC_	MPE	G2_SLI	CEGRO	UP_S	TATE	
Source:		VideoCS						
Length E	Bias:	2						
This is a multiple	slice gr slice g	roup level command roups. The same co	d and can mmand is	be issued n used for A	nultiple times VC encoder (P/	within a AK mode	picture that i e) and decod	s comprised of er (VLD and IT
DWord	Bit				Description	n		
0	31:29	Command Type						
-		Default Value:		3h F	PARALLEL_VIDE	EO_PIPE		
		Format:		OpC	Code			
	28:27	Pipeline						
		Default Value:		2h MFX_MF	PEG2_SLICEGR	OUP_STA	ATE	
		Format:		OpCode				
	26:24	Media Command	Opcode					
		Default Value:	•			3h MP	EG2	
		Format:				OpCoc	le	
	23:21	SubOpcode A				•		
		Default Value:				2h MED	IA_	
		Format:				OpCode	9	
	20:16	SubOpcode B						
		Default Value:				3h MED	IA_	
		Format:				OpCode	9	
	15:12	Reserved						
		Format:				MB	Z	
	11:0	DWord Length						
		Default Value:		6h	Excludes DWc	ord (0,1)		
		Format:		=n	Total Length ·	- 2		
1	31	MbRateCtrlFlag-	RateConti	rolCounter	Enable (Encoc	der-only)	
		To enable the accu Control logic. The hardware ignores t bits MbRateCtrlFla Value 0h	imulation rest of the hese field: g and Mbl	of bit alloca RC control s. Note: To RateCtrlRes Disable	ition for rate of fields are only reset MB level et to 1 in the r Name	ontrolTh valid wh rate con new slice	is field enabl nen this field trol (QRC), w	es hardware Rate is set to 1. Otherwise, re need to set both
	20	MbBataCtriBasat	PocotP-4	-nabic	ountor (Enco	dor only	<u> </u>	
	30	To reset the bit allo	ocation ac	cumulation	counter to 0 t	o restart) the rate con	trol.
		Value	Na	ame	De	escriptio	n	
		0h	Disable		Not reset			
		1h	Enable		reset			



29:28	9:28 MbRateCtrlMode- RC Triggle Mode (Encoder-only)								
	Value Name Description								
	00b	Always Rat sum_act <	e Control, whereas RC sum_target	becomes acti	ve if sum_act > sum	n_target or			
	01b	Gentle Rate sum_act <	Gentle Rate Control, whereas RC becomes active if sum_act > upper_midpt or um_act < lower_midpt						
	10b	Loose Rate sum_act <	Control, whereas RC b sum_min	pecomes activ	e if sum_act > sum_	_max or			
	11b	Reserved							
27:24	MbRateCtr	MbRateCtrlParam- RC Stable Tolerance (Encoder-only)							
	Format: U4								
	This field sp	ecifies the tole	rance required to deac	tivate RC onc	e it has been trigge	red.			
		Value	9		Name				
	[0, 15]								
	controls wh	at type of panie Value	c behavior is invoked. Name	behavior is invoked. Name					
	0		Disable	Disable					
	-								
22			Enable						
22	Image: Image shows a straight of the second straight of the	nicType - RC P lects between t d out, setting to , AC CBPs are s CBPs are forced alue	Enable anic Type (Encoder-o two RC Panic methods. o requested QP + QP_r et to zero (note that D to zero. Name	only) . If it is set to (max_pos_mod C CBPs are no QP Panic), in panic mode, th . If it is set to 1, for ot modified). For inte Description	e macrobloc an intra er macroblo			
22	Image:	nicType - RC P lects between t d out, setting to , AC CBPs are s CBPs are forced hlue	Enable anic Type (Encoder-o two RC Panic methods. o requested QP + QP_r et to zero (note that D to zero. Name	nly) . If it is set to (max_pos_mod C CBPs are no QP Panic CBP Panic	0, in panic mode, th . If it is set to 1, for ot modified). For inte Description	e macrobloc an intra er macroblo			
22	Image: Constraint of the second se	nicType - RC P lects between t d out, setting to , AC CBPs are s CBPs are forced hlue	Enable anic Type (Encoder-o two RC Panic methods. o requested QP + QP_r et to zero (note that D d to zero. Name	nly) . If it is set to (max_pos_mod C CBPs are no QP Panic CBP Panic), in panic mode, th . If it is set to 1, for ot modified). For inte Description	e macrobloc an intra er macroblo			
22 21	Image: Image shows a straight of the straightof to the straight of the straight of the straight of the straight	nicType - RC P lects between t d out, setting to , AC CBPs are s CBPs are forced alue	Enable anic Type (Encoder-o two RC Panic methods. o requested QP + QP_r et to zero (note that D d to zero. Name	nly) If it is set to (max_pos_mod C CBPs are no QP Panic CBP Panic MB), in panic mode, th . If it is set to 1, for ot modified). For inte Description Z	e macrobloo an intra er macroblo			
22 21 20	Image: Image shows a straight of the straightos straightos straight of the straight of the straight of the stra	nicType - RC P lects between t d out, setting to , AC CBPs are s CBPs are forced alue isabled - MB T only valid for a 3.1.6	Enable anic Type (Encoder-o two RC Panic methods. o requested QP + QP_r et to zero (note that D d to zero. Name Type Skip Conversion P or B slice. It must be	nly) If it is set to the max_pos_mod C CBPs are not QP Panic CBP Panic CBP Panic MB Disable (Encestion of the first set the first set to the first set t	0, in panic mode, th . If it is set to 1, for ot modified). For inte Description Z oder-only) er slice types. Rules	e macrobloo an intra er macroblo are provideo			
22 21 20	Image: Image shows a straight of the straightos straightos straight of the straight of the straight of the stra	hicType - RC P lects between t d out, setting to , AC CBPs are s CBPs are forced alue isabled - MB T only valid for a 3.1.6 Name	Enable anic Type (Encoder-o two RC Panic methods. o requested QP + QP_r et to zero (note that D d to zero. Name Type Skip Conversion P or B slice. It must be	pnly) . If it is set to 0 max_pos_mod C CBPs are no QP Panic CBP Panic MB Disable (Enc e zero for othe Description	2, in panic mode, th . If it is set to 1, for ot modified). For inte Description Z oder-only) er slice types. Rules	e macrobloc an intra er macroblo are providec			
22 21 20	Image:	hicType - RC P lects between t d out, setting to , AC CBPs are s CBPs are forced hlue isabled - MB T only valid for a 3.1.6 Name Enable	Enable anic Type (Encoder-o wo RC Panic methods. requested QP + QP_r et to zero (note that D to zero. Name ype Skip Conversion P or B slice. It must be Enable skip type con	If it is set to 0 max_pos_mod C CBPs are no QP Panic CBP Panic MB Disable (Encemption of the provided of the provid	0, in panic mode, th . If it is set to 1, for ot modified). For inte Description Z oder-only) er slice types. Rules	e macrobloc an intra er macroblo are providec			
22 21 20	Image:	hicType - RC P lects between t d out, setting to , AC CBPs are s CBPs are forced hlue isabled - MB T only valid for a 3.1.6 Name Enable Disable	Enable anic Type (Encoder-o two RC Panic methods. two RC Panic methods. to requested QP + QP_r et to zero (note that D to zero. Name Type Skip Conversion P or B slice. It must be Enable skip type co	nly) . If it is set to on max_pos_mod C CBPs are not QP Panic CBP Panic CBP Panic MB Disable (Encent e zero for other Description nversion	0, in panic mode, th . If it is set to 1, for ot modified). For inte Description Z oder-only) er slice types. Rules	e macroblog an intra er macroblo are provideo			
22 21 20 19	Image: Constraint of the sector of the se	hicType - RC P lects between t d out, setting to , AC CBPs are s CBPs are forced hlue isabled - MB T only valid for a 3.1.6 Enable Disable Grp rp = 1 if the cu	Enable anic Type (Encoder-o swo RC Panic methods. b requested QP + QP_r et to zero (note that D d to zero. Name Type Skip Conversion P or B slice. It must be Enable skip type con Disable skip type con rrent slice group is the he Minimum Frame Si	If it is set to the max_pos_mod C CBPs are not QP Panic CBP Panic CBP Panic MB Disable (Enc e zero for othe Description nversion	2 D, in panic mode, th . If it is set to 1, for bt modified). For inter- Description Z oder-only) er slice types. Rules of up of a picture; 0 ot	e macroblo an intra er macroblo are provideo herwise. It is			

			MFC	_MPEG2_SLICEGROUP_STATE						
	17	HeaderPresentFlag - Header Insertion Present in Bitstream (Encoder-only)								
		Value Name		Description						
		0h	Disable	no header insertion into the output bitstream buffer, in front of the current slice encoded bits						
		1h	Enable	header insertion into the output bitstream buffer is present, and is in front of the current slice encoded bits.						
	16	SliceData PresentFlag - SliceData Insertion Present in Bitstream (Encoder-only)								
		Value	Name	Description						
		0h	Disable	no Slice Data insertion into the output bitstream buffer						
		1h	Enable	Slice Data insertion into the output bitstream buffer is present.						
	15	TailPresentFlag - Tail Insertion Present in bitstream (Encoder-only)								
		Value	Name	Description						
		0h	r	no tail insertion into the output bitstream buffer, after the current slice encoded bits						
		1h	t	ail insertion into the output bitstream buffer is present, and is after the current lice encoded bits.						
	14	FirstSliceHdrDisabled when this is on, the first slice header of the slice group is expected to be provided by the user via insertion command. PAK HW will skip it.								
	13	IntraSlice intra slice value included in slice headers, when IntraSliceFlag = 1.								
	12	IntraSliceFlag intra slice flag included in slice headers								
	11:8	Reserved								
		Format		MBZ for SliceID extension						
	7:4	 4 SliceID[3:0] (Encoder-only) To identify the output data (coding information record) returned for rate control from PAK ENC and VPP 								
	3:2	Reserved								
		Format		MBZ for StreamID extension						
	1:0	StreamID[1:0] (Encoder-only) To identify the output data (coding information record) returned for rate control from PAK to ENC and VPP								
2	31:24	NextSgMbYcnt - also NextStartVertPos Vertical count of the first MB in the next slice group (Encoder-only)Note: This field restricts total number of MB in the Y direction to 255 or less.								
	23:16	NextSgMbXcnt - also NextStartHorzPos BitFieldDesc								
	15:8	FirstMbYcnt - also CurrStartVertPos								
		Format: U8								
		also CurrStartVertPos, Vertical count of the first MB in the current slice group (Encoder-only)								



MFC_MPEG2_SLICEGROUP_STATE

-										
	7:0	FirstMbXcnt - also CurrStartHorzPos								
		Format: U8								
		Horizontal count of the first MB in the current slice group (Encoder-only)								
3	31:9	Reserved								
		Format:	MBZ							
	8	SliceGroupSkip								
		Exists If: //Encoder Only								
		Format: U1								
		All macroblocks are skipped								
	7:6	Reserved								
		Format:				MBZ				
	5:0	SliceGroupOp								
		Exists If: //Encoder Only								
		Format: U6								
		Initial slice quality parameter								
4	21.20									
4	51.29	Format:		MB7						
	28.0	RitstreamOffset - Indirect BAK-BSE Data Start Address (W/rite)								
	20.0	Exists If: //Encoder Only								
		This field specifies the memory starting address (offset) to write out the compressed bitstream data from the BSE processing. This pointer is relative to the MFC Indirect PAK-BSE Object Base Address. It is a byte-aligned address for the AVC bitstream data in both CABAC/CAVLC Modes. For Write, there is no need to have a data length field. It is assumed the global memory bound check specified in the IND_OBJ_BASE_ADDRESS command (Indirect PAK-BSE Object Access Upper Bound) will take care of any illegal write access. This field is only valid for AVC encode mode.								
		Value				Name				
		[0,512MB)								
5	31:24	MaxQpNegModifier - Magnitude of QP Max Negative Modifier (Encoder-only)								
		Format:		U8						
		This field specifies the lower limit of the QP modifier.								
		Value	Name							
		[0, 51]								
	23:16	MaxQpPosModifier - Magnitude of QP Max Positive Modifier (Encoder-only)								
		Format: U8								
		This field specifies the upper limit of the QP modifier.								


		MFC_MPEG2_SLICEG	ROUP_STATE							
		Value	Name							
		[0, 51]								
	15:12	ShrinkParam - Shrink Resistance (Encoder-only)								
		Format:	U4							
		This field specifies the additional points added each time decreased correction is invoked.								
		Value	Name							
		[0, 15]								
	11:8	Shrinkaram - Shrink Init (Encoder-only)								
		Format:	U4							
		This field specifies the initial points required to t	rip decreased control.							
		Value	Name							
		[0, 15]								
	7:4	GrowParam - Grow Resistance (Encoder-only))							
		Format:	U4							
		This field specifies the additional points added e	ach time increased correction is invoked.							
		Value	Name							
		[0, 15]								
	3:0	GrowParam - Grow Init (Encoder-only)								
		Format: U4								
		This field specifies the initial points required to trip increased control.								
		Value	Name							
		[0, 15]								
6	31:24	Reserved								
		Format:	MBZ							
	23:20	CorrectPoints - Correct 6 (Encoder-only)								
		Format:	U4							
		This field specifies the points used in the lowern	nost RC region when sum_act <= sum_min.							
		Value	Name							
		[0, 15]								
	19:16	CorrectPoints - Correct 5 (Encoder-only)								
		Format:	U4							
		This field specifies the points used in the fifth R	C region when sum_act > sum_min but <=							
		lower_midpt.								
		Value	Name							
		[0, 15]								
	15:12	CorrectPoints - Correct 4 (Encoder-only)								
		Format:	U4							
		This field specifies the points used in the fourth	RC region when sum_act > lower_midpt but <=							
		sum_target.								



MFC_MPEG2_SLICEGROUP_STATE

		Value			Name					
		[0, 15]								
	11:8	CorrectPoints - Correct 3 (Encoder-only)								
		Format:			U4					
		This field specifies the points used	in the third R	C region when sur	m_act > sum_target but <=					
		Value			Name					
		[0, 15]								
	7.4	CorrectPoints - Correct 2 (Encode	er-only)							
	7.7	Format:	er-only)		U4					
		This field specifies the points used <= sum_max.	in the second	d RC region when s	sum_act > upper_midpt but					
		Value			Name					
		[0, 15]								
	3:0	CorrectPoints - Correct 1 (Encode	er-only)							
		Format:	-		U4					
		This field specifies the points used	st RC region wher	i sum_act > sum_max						
		Value			Name					
		[0, 15]								
7	31:28	CV7 - Clamp Value 7 (Encoder-only)								
		Exists If: //Encoder Only								
	27:24	CV6 - Clamp Value 6 (Encoder-only)								
		Exists If:	//Encoder O	nly						
		Format:	U4							
	23:20	CV5 - Clamp Value 5 (Encoder-or	nly)							
		Exists If:	//Encoder O	nly						
		Format:	U4							
	19:16	CV4 - Clamp Value 4 (Encoder-or	nly)							
		Exists If:	//Encoder O	nly						
		Format:	U4							
	15:12	CV3 - Clamp Value 3 (Encoder-or	nly)							
		Exists If:	//Encoder O	nly						
		Format:	U4							
	11:8	CV2 - Clamp Value 2 (Encoder-or	nly)							
		Exists If:	//Encoder O	nly						
		Format:	U4							
	7:4	CV1 - Clamp Value 1 (Encoder-or	nly)							
		Exists If:	//Encoder O	nly						



MFC_MPEG2_SLICEGROUP_STATE

	Forma	at:					U4			
3:0	CV0 -	CV0 - Clamp Value 0 (Encoder-only)								
	If the	magni	itude	of co	peffic	ients	at lo	catio	ns assigned with CV0 (mapping shown below)	
	excee	ds 2C	/0-1,	they	are r	eplac	ed w	ith 2	CV0-1. For coefficients at locations marked as	
	'none'	, no c	lamp	ing is	perf	orme	ed. Th	ne fol	lowing mappings are only applied to luma and	
	chrom		CKS\SI		CKS (conta	ining	AC	coefficiencts (blocks/sublocks with only DC	
	For 8v	WIII N 8 fran			ipeu, ach i). Coeffi	iciont	· ic m	anned to one of the eight CV values as	
	follow	ina:		JCK , θ	acri	COEIII	Clerit	. 15 111	apped to one of the eight CV values as	
	none	none	CV7	CV6	CV5	CV4	CV3	CV3		
	none	CV7	CV6	CV5	CV4	CV3	CV3	CV2		
	CV7	CV6	CV5	CV4	CV3	CV3	CV2	CV2		
	CV6	CV5	CV4	CV3	CV3	CV2	CV2	CV1		
	CV5	CV4	CV3	CV3	CV2	CV2	CV1	CV1		
	CV4	CV3	CV3	CV2	CV2	CV1	CV1	CV0		
	CV3	CV3	CV2	CV2	CV1	CV1	CV0	CV0		
	CV3	CV2	CV2	CV1	CV1	CV0	CV0	CV0		
	For 8x	8 field	d bloo	ck, ea	ch co	beffic	ient i	s ma	pped to one of the eight CV values as	
	follow	ving:	1		-			-		
	none	none	CV6	CV5	CV4	CV3	CV2	CV1		
	none	CV7	CV6	CV5	CV4	CV3	CV2	CV1		
	CV7	CV6	CV5	CV4	CV3	CV3	CV2	CV1		
	CV7	CV6	CV5	CV4	CV3	CV2	CV2	CV1		
	CV6	CV5	CV4	CV4	CV3	CV2	CV1	CV0		
	CV6	CV5	CV4	CV3	CV2	CV2	CV1	CV0		
	CV5	CV5	CV4	CV3	CV2	CV1	CV1	CV0		
	CV5	CV5	CV4	CV3	CV2	CV1	CV1	CV0		



MFD_AVC_BSD_OBJECT

Source: VideoCS

Length Bias:

The MFD_AVC_BSD_OBJECT command is the only primitive command for the AVC Decoding Pipeline. The same command is used for both CABAC and CAVLD modes.

The Slice Data portion of the bitstream is loaded as indirect data object.Before issuing a MFD_AVC_BSD_OBJECT command, all AVC states of the MFD Engine need to be valid. Therefore the commands used to set these states need to have been issued prior to the issue of a MFD_AVC_BSD_OBJECT command.

DWord	Bit	Description						
0	31:29	Command Type						
		Default Value:	3h PARALLEL_\	/IDEO_PI	PE			
		Format:	OpCode					
	28:27	Pipeline						
		Default Value:	2h MFD_AVC_B	SD_OBJE	СТ			
		Format:	OpCode					
-	26:24	Media Command Opcode						
		Default Value:		1h AVC	_DEC			
		Format:		OpCode	e .			
	23:21	SubOpcode A						
		Default Value:			1h			
		Format:	OpCode					
2	20:16	SubOpcode B						
		Default Value:			8h			
		Format:			OpCode			
	15:12	Reserved						
		Format:			MBZ			
	11:0	DWord Length						
		Default Value:	4h Excludes DWord	(0,1) = 0	0004			
		Format:	=n Total Length - 2					
1	31:24	Reserved						
		Format:			MBZ			
	23:0	Indirect BSD Data Length						
		Format:			U24			
		This field provides the length i data fetching is disabled - sub This field must have the same AVC Short Format: It is the len Slice Header + Slice Data + En MB.	in bytes of the indire sequently, the Indire a alignment as the In ngth in bytes of the I nulation Prevention	ct data. A ect Data S direct Ok bitstream Bytes + a	A value zero indicates that indirect Start Address field is ignored. Diect Data Start Address. In data for the current slice, including any filling trailing zeros after the last			



		MFD_AVC_BSD_OBJE	СТ				
		Hardware ignores the contents after the last non-zero byte. Trailing zero is allowed and handled correctly in both CABAC and CAVLC modes.					
2	31:29	Reserved					
		Format:	MBZ				
	28:0	Indirect BSD Data Start Address This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This pointer is relative to the MFD Indirect Object Base Address . Hardware ignores this field if indirect data is not present. It is a byte-aligned address for the AVC bitstream data in both CABAC/CAVLD Modes. In implementing a phantom slice at the end of a picture for automatic error concealment, this field should set to 0. It includes the NAL Header (the NAL Header does not need to perform EMU detection). For AVC Base Layer, it is a single byte. But for MVC, the NAL Header is 4 Bytes long. These NAL Header Unit must be passed to HW in the compressed bitstream buffer.					
		[0,512MB)					
35	31:0	Inline Data All the required Slice Header parameters and error handlir of the AVC_BSD_OBJECT command. It has a fixed size of 4 follwoing section: Inline Data Description.	ng settings are captured as InLine Data DWs. Its definition is described in the				



			MFD_AVC	_DPB_S	ΓΑΤ	E	
Source:		VideoC	S				
Length E	Bias:	2					
This is a RefFram MFX_PIP collected FieldOrd POCList[frame l neList[1 PE_BUF_ d into L derCntL 34] of l	evel state com 6] of DXVA2 ir ADDR_STATE ongTermPic_F .ist[16][2] and MFX_AVC_DIR	nmand used only in DXV nterface is replaced with command. The LongTerr lag[16]. CurrFieldOrderCnt[2] of ECTMODE_STATE comma	A2 AVC Short S this implement m Picture flag ir DXVA2 interfac and.	lice Bit ation's ndicato e are re	stream Format VLD Reference Picture A r of all reference pic eplaced with this imp	mode. ddresses[16] of tures are plementation's
DWord	Bit			Descriptio	on		
0	31:29	Command Ty	/ре				
		Default Value	e: 3h	n PARALLEL_VIC	DEO_PI	PE	
		Format:	O	pCode			
	28:27	Pipeline					
		Default Value	2:	2h MFX_MU	JLTI_D\	N	
		Format:		OpCode			
	26:24	Media Comm	and Opcode				
		Default Value	fault Value: 1h A			DEC	
2		Format:		C	DpCode	2	
	23:21	SubOpcode A	4				
		Default Value	fault Value:			1h	
		Format:				OpCode	
	20:16	SubOpcode B					
		Default Value	2:			6h	
		Format:			OpCode		
	15:12	Reserved					
		Format:			١	MBZ	
	11:0	DWord Leng	th				
		Format:	=n Tota	l Length - 2			
		Value		Name			
		9h	Excludes DWord (0,1)	[Default]			
1	31:16	LongTermFra	me_Flag[16][1 bit]				
		One-to-one c	orrespondence with the	entries of the tl	his imp	lementation's RefFra	ameList[16]. 1 bit
		per reference	frame.				
		Value		1	Name		
		1	the picture is a long ter	m reterence pic	ture		
		0	the picture is a short ter	rm reference pi	cture		
	15:0	Non-Existing	Frame_Flag[16][1 bit]				
		Une-to-one c	orrespondence with the	entries of the t	nis imp	ementation's RefFra	imeList[16]. 1 bit



MFD_AVC_DPB_STATE

		per refe	rence fram	ne.			
		Value	lue Name Description				
	1 INVALID the reference picture in that entry of RefFram			picture in that entry of RefFrameList[] does not exist anymore.			
		0	VALID	the refe	erence	picture in that entry of RefFrameList[] is a valid reference	
						Programming Notes	
		When a entry b bit of N	an elemen [.] eing empt NonExisting	t of the y or bei gFrameF	list of f ng mar lags sh	rames is not relevant (e.g., due to the corresponding reference ked as "not used for reference"), the value of the corresponding all be set to 0.	
2	31:0	UsedFo	rReferenc	e Flag[16][2 k	pits]	
		One-to- per refe	one corre	sponder ne.	nce with	h the entries of the this implementation's RefFrameList[16]. 2 bits	
		Value	e Name Description			Description	
		0	NOT_REFE	RENCE	indica	tes a frame is "not used for reference".	
		1	TOP_FIELD	0	bit[0] refere	indicates that the top field of a frame is marked as "used for nce".	
		2	BOTTOM_	FIELD	bit[1] refere	indicates that the bottom field of a frame is marked as "used for nce".	
		3	FRAME		bit[1:0 refere] indicates that a frame (or field pair) is marked as "used for nce".	
310	31:0	LTSTFra	ameNumL	.ist[16][16 bits		
		One-to-	one corre	sponder	nce witl	h the entries of the this implementation's RefFrameList[16]. 16	
		bits per	reference	frame. [Depend	ling on the corresponding LongTermFrame_Flag[], the content of	
		Value		lame	ierentiy	Description	
		1	l ongTerm	Frame	Flaglil	I TSTErameNuml ist[i] represent LongTermErameIdy	
		0	LongTerm			I TSTFrameNumListfilrepresent Short Term Picture FrameNum	
		Ŭ					
			Programming Notes				
		When a entry b LTSTFr	an elemen [.] eing empt ameNumLi	t of the y or bei ist entry	list of f ng mar shall b	rames is not relevant (e.g., due to the corresponding reference ked as "not used for reference"), the value of the e set to 0.	



MFD_AVC_SLICEADDR

Source: VideoCS Length Bias: 2

This is a Slice level command used only for DXVA2 AVC Short Slice Bitstream Format VLD mode. When decoding a slice, H/W needs to know the last MB of the slice has reached in order to start decoding the next slice. It also needs to know if a slice is terminated but the last MB has not reached, error conealment should be invoked to generate those missing MBs. For AVC DXVA2 Short Format, the only way to know the last MB position of the current slice, H/W needs to snoop into the next slice's start MB address (a linear address encoded in the Slice Header). Since each BSD Object command can have only one indirect bitstream buffer address, this command is added to help H/W to snoop into the next slice's slice header and retrieve its Start MB Address. This command will take the next slice's bitstream buffer address as input (exactly the same way as a BSD Object command), and parse only the first_mb_in_slice syntax element. The result will stored inside the H/W, and will be used to decode the current slice specified in the BSD Object command. Only the very first few bytes (max 5 bytes for a max 4K picture) of the Slice Header will be decoded, the rest of the bitstream are don't care. This is because the first_mb_in_slice is encoded in Exponential Golomb, and will take 33 bits to represent the max 256 x 256 = 64K-1 value. The indirect data of MFD_AVC_SLICEADDR is a valid BSD object and is decoded as in BSD OBJECT command. The next Slice Start MB Address is also exposed to the MMIO interface. The Slice Start MB Address (first_mb_in_slice) is a linear MB address count; but it is translated into the corresponding 2D MB X and Y raster position, and are stored internally as NextSliceMbY and NextSliceMbX.

DWord	Bit	Description							
0	31:29	Command Type							
		Default Value: 3h PARALLEL_VIDEO_			IPE				
		Format:	OpCode						
	28:27	Pipeline							
		Default Value: 2h MFD_AVC_ SLICEA			DR				
		Format: OpCode							
	26:24	Media Command Opcode							
		Default Value:			_DEC				
		Format:			OpCode				
	23:21	SubOpcode A							
		Default Value:			1h				
		Format:			OpCode				
	20:16	SubOpcode B							
		Default Value:			7h				
		Format:			OpCode				
	15:12	Reserved							
		Format:			MBZ				
	11:0	DWord Length							
		Default Value:	1h Excludes DV	Vord (0	,1)				
		Format:	=n Total Lengt	h - 2					
1	31:24	Reserved							
		Format:			MBZ				



		MFD_AVC_S	SLICEADD	DR		
	23:0	Indirect BSD Data Length				
		Format: U	24 in bytes			
		This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Data Start Address field is ignored. Driver always programs this up to 5 bytes; for bitstream less than 5 bytes, driver program the lesser value. (Emulation Prevention Byte should never happen for the first 5 bytes when the max picture size can only be 4Kx4K)It is the length in bytes of the bitstream data for the current slice, including Slice Header + Slice Data + Emulation Prevention Bytes + any filling trailing zeros aft the last MB. Hardware ignores the contents after the last non-zero byte. Trailing zero is allowed and handled correctly in both CABAC and CAVLC modes.				
2	31:29	Reserved				
		Format:		MBZ		
	28:0	Indirect BSD Data Start Address This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This pointer is relative to the MFD Indirect Object Base Address. Hardware ignores this field if indirect data is not present. It is a byte-aligned address for the AVC bitstream data in both CABAC/CAVLD Modes. In implementing a phantom slice at the end of a picture for automatic error concealment, this field should set to 0.It includes the NAL Header Byte. (but does not perform EMU detection). Must provide a valid MB address, even if error. MB must be clamped to within a pic boundary.				
		Value		Name		
		[0,512MB)				

Г



			MFD_	IT_OBJE	ECT				
Source:		Video	cs						
Length E	Length Bias: 2								
All weigl implicit.	nt mod	e (default and	d implicit) are mapped	to explicit mod	le. But the weights come in either as explicit c				
DWord	Bit			Descrip	otion				
0	31:29	Command T	Гуре						
		Default Valu	ie:	3h PARALLEL_	VIDEO_PIPE				
		Format:		OpCode					
	28:27	Pipeline							
		Default Valu	le:	2h MFD	_IT_OBJECT				
		Format:		OpCode	e				
	26:24	Media Com	mand Opcode						
		Default Valu	le:	0h MFX_COM	/MON_DEC				
		Format:		OpCode					
	23:21	SubOpcode	Α						
		Default Valu	le:	lh					
		Format:			OpCode				
	20:16	SubOpcode	В						
		Default Valu	le:		9h				
		Format:			OpCode				
	15:12	Reserved							
		Format:			MBZ				
	11:0	DWord Length							
		Default Value:	Default 06h Excludes DWord (0,1) For AVC = Ch Value:						
		Format:	=n Total Length - 2 I this command.	Note: Regardle	ess of the mode, inline data must be present in				
1	31:10	Reserved							
		Format:			MBZ				
	9:0	Indirect IT-N	MV Data Length						
		Format:	U10 Forma	atDesc: In bytes	S				
		This field pro	This field provides the length in bytes of the indirect data, which contains all the MVs for the						
		current MB (in any partitioning and	subpartitioning	g form). A value zero indicates that indirect				
		This field mu	ist have the same align	ment as the India	direct Object Data Start Address. AVC-IT Mod				
		It must be D	Word aligned (since ea	ch MV is 4byte	s in size) Driver has to derived this field from				
		MVsize (MVc	quantity in DXVA, exact	: size) *4 bytes	per MV. This field is only valid in AVC decode				
		ll mode ////	1 and MDEC uses inline	M(A + a)					
		II mode (VC	1 and MPEG uses inline	e MV data).					



		Μ	FD_IT_OBJECT				
		Format:		MBZ			
	28:0	Indirect IT-MV Data Start Address Offset This field specifies the memory starting address (offset) of the MV data to be fetched into the IT pipeline for processing. This pointer is relative to the Indirect IT-MV Object Base Address. Hardware ignores this field if indirect data is not present, i.e. the Indirect MV Data Length is set to 0. Alignment of this address depends on the mode of operation. AVC-IT Mode: It must be DWord aligned (since each MV is 4 bytes in size). This field is only valid in AVC decoder IT mode (VC1 and MPEG uses inline MV data).					
		Val	ue	Name			
		[0,512MB)					
3	31:12	Reserved					
		Format:		MBZ			
	11:0	Indirect IT-COEFF Data Leng This field provides the length coefficients for the current MI subsequently, the Indirect IT-6 is 1 DW in size, with 12 bits, th 3 byte pixel components * 4 b AVC (since each coefficient is IT mode.	Ith in bytes of the indirect data, B. A value zero indicates that COEFF Data Start Address fie his field can be extended to s bytes per coeff). This field mu 4 bytes in size). This field is c	which contains all the non-zero indirect data fetching is disabled - ld is ignored. Since each IT-COEFF data support up to 4:4:4 format.(256 pixel * st be integer multiple of 16-bytes for only valid in AVC, VC1, MPEG2 decoder			
		Value		Name			
		[0,3072]	In bytes [0, 256*3*4]				
4	31:29	Reserved					
		Format:		MBZ			
	28:0	Indirect IT-COEFF Data Start This field specifies the memor pipeline for processing. This p Hardware ignores this field if Length is set to 0.This field me will determine the Num of EO cannot hang - add error hand Val	Indirect IT-COEFF Data Start Address Offset This field specifies the memory starting address (offset) of the coeff data to be loaded into the IT pipeline for processing. This pointer is relative to the Indirect IT-COEFF Object Base Address. Hardware ignores this field if indirect IT-COEFF data is not present, i.e. the Indirect IT-COEFF Data Length is set to 0.This field must be DW aligned, since each coeff icient is 4 bytes in size. Driver will determine the Num of EOB 4x4/8x8 must match the block cbp flags, if not match, hardware cannot hang - add error handling. This field is only valid in AVC, VC1, MPEG2 decoder IT mode.				
		[0,512MB)					
5	31:6	Reserved					
		Format:		MBZ			
	5:0	Indirect IT-DBLK Control Da	ta Length				
		Format:		U6			
		This field provides the length control information for the cu indirect data fetching is disab ignored. This field must have must be DWord aligned. Each	in bytes of the indirect data, rrent MB (in 4x4 sub-block p led - subsequently, the Indire the same alignment as the In Deblock Control Data record	which contains all the deblocker artitioning). A value zero indicates that ect IT-DBLK Data Start Address field is idirect IT-DBLK Data Start Address. It d is 48 bytes or 12 DWords in size. This			



			MFD_IT_OBJECT					
		field is only valid in	AVC decoder IT mode.					
6	31:29	Reserved						
		Format:		MBZ				
	28:0	Indirect IT-DBLK C	ontrol Data Start Address Offset					
		Format:	IndirectObjectBaseAddress[28:0]					
		This field specifies the fetched into the IT P Base Address. Hardware ignores t Length is set to 0. It must be DWord a This field is only val	This field specifies the memory starting address (offset) of the Deblocker control data to be fetched into the IT Pipeline for processing. This pointer is relative to the Indirect IT-DBLK Object Base Address. Hardware ignores this field if indirect data is not present, ie. The indirect IT-DBLK Control Data Length is set to 0. It must be DWord aligned. Each Deblock Control Data record is 48 bytes or 12 DWords in size. This field is only valid in AVC decoder IT mode					
			Value	Name				
		[0,512MB)						
7n	31:0	Inline Data Union for all 3 code Includes IT, MC, Int AVC-IT Modes: Har VC1-IT Modes: Har MPEG2-IT Modes: H For AVC there 7 DV	cs raPred inline data as well as Deblocker dware interprets this data in the specif dware interprets this data in the specif fardware interprets this data in the spe fords of inline data, hence N is equal t	control information ied format. ied format. MV inline ecified format. (IS mode) MV inline o 13.				



		MFD_JF	PEG_BSD_O	BJECT
Source:		VideoCS		
Length E	Bias:	2		
Exists If:		//Decoder		
DWord	Bit		Descripti	on
0	31:29	Command Type		
		Default Value:	3h PARALLEL_VI	DEO_PIPE
		Format:	OpCode	
	28:27	Pipeline		
		Default Value:	2h MFD_JPEG_BS	D_OBJECT
		Format:	OpCode	
	26:24	Media Command Opcode		
		Default Value:	7	h JPEG_DEC
		Format:	С	0pCode
	23:21	SubOpcode A		
		Default Value:		1h
		Format:		OpCode
	20:16	SubOpcode B		
		Default Value:		8h
		Format:		OpCode
	15:12	Reserved		
		Format:		MBZ
	11:0	DWord Length		
		Default Value:	004h Excludes D\	Vord (0,1)
		Format:	=n Total Length -	- 2
1	31:0	Indirect Data Length . It is the length in bytes of the last first MCU and the last non-zero bytes (if present) are excluded.	oitstream data for the byte of the last MCL Hardware ignores the	e current Scan. It includes the first byte of the J in the Scan. Specifically, the zero-padding e contents after the last non-zero byte.
2	31:29	Reserved		
		Format:		MBZ
	28:0	Indirect Data Start Address This field specifies the Graphics for processing. This pointer is re this field if indirect data is not p	Memory starting add alative to the BSD Incorresent. It is a byte-al	dress of the data to be fetched into BSD Unit lirect Object Base Address. Hardware ignores igned address for the JPEG bitstream data
3	31:29	Reserved		
		Format:		MBZ
	28:16	Scan Horizontal Position		
		Format:	U13 bits in blocks	
		This field indicates the horizonta	al position (in block ι	units) of the first MCU in the Scan.



MFD_JPEG_BSD_OBJECT

	1 - 1 -	<u> </u>								
	15:13	Reserved		NAD 7						
		Format:			MBZ					
	12:0	Scan Vertical Position								
		Format:	U13 bits	in blocks						
		This field in	dicates the vertical position (in block units) of t	he first MCU in the Scan.					
4	31	Reserved								
		Format:			MBZ					
	30	Interleaved	1	_						
		Value	Name		Description					
		0	Non-Interleaved	one component	in the Scan					
		1	Interleaved	multiple compo	onents in the Scan					
		Bit0: Y Bit1: U Bit2: V For exampl set to 111b	le, if non-interleaved Y, then	it will be set to 00:	1b. If interleaved Y, U, and V, it will be					
	26	Reserved								
		Format:			MBZ					
	25:0	MCU Count								
		Format:			U26					
		This field indicates the number of MCUs in the Scan.								
5	31:16	Reserved								
		Format:			MBZ					
	15:0	RestartInte	erval(16 bit)							
		Format:			U16					
		Specifies th that all the	Specifies the number of MCU in restart interval. Valid values are 1->0xFFFFValue of 0 implies that all the SCAN have only one ECS.							



MFD_MPEG2_BSD_OBJECT

Source: VideoCS

2

Length Bias:

Different from AVC and VC1, MFD_MPEG2_BSD_OBJECT command is pipelinable. This is for performance purpose as in MPEG2 a slice is defined as a group of MBs of any size that must be within a macroblock row. Slice header parameters are passed in as inline data and the bitstream data for the slice is passed in as indirect data. Of the inline data, slice_horizontal_position and slice_vertical_position determines the location within the destination picture of the first macroblock in the slice. The content in this command is identical to that in the MEDIA_OBJECT command in VLD mode described in the Media Chapter.

DWord	Bit		Description						
0	31:29	Command Type							
		Default Value:	3h PARALL	.EL_VIDEO_PI	PE				
		Format:	OpCode						
	28:27	Pipeline							
		Default Value:	2h MFD_MPE	G2_BSD_OBJ	ECT				
		Format:	OpCode						
	26:24	Media Command Opcode							
		Default Value:		3h MPEG2_I	DEC				
		Format:		OpCode					
	23:21	SubOpcode A							
		Default Value:			1h				
		Format:			OpCode				
2	20:16	SubOpcode B							
		Default Value:			8h				
		Format:			OpCode				
	15:12	Reserved							
		Format:		MBZ					
	11:0	DWord Length							
		Default Value:	des DWord ((0,1)					
		Format:	=n Total Len	igth - 2					
1	31:0	Indirect BSD Data Length							
		Format:			U32				
		It is the length in bytes of the bitstream data for the current slice. It includes the first byte of the							
		first macroblock and the last non-zero byte of the last macroblock in the slice. Specifically, the							
		zero-padding bytes (if present) and the next start-code are excluded. This field is sized to support beyond MPEG-2 MP@HL bitstream ($<4K$). According to Table 8-6 of							
	ISO/IEC 13818-2, the maximum number of bits per macroblock for 4:2:0 is 4608. S								
		maximum slice size for 4K x 4K is 4608 * 256 / 8 = 147,456 bytes (0x24000), which requ							
			Programmin	g Notes					
		As MPEG-2 spec does not post a	any limitation	of the size of	f zero-padding bytes, it is				



		MFD_MPEG2_BSD_OBJECT	
		possible to have a slice data with large length (including zero-padding bytes). As the data beyond 0x10E00 would only be zero bytes for a valid slice data	
		Hardware does not handle zero-padding at the end of the slice data so driver needs to program the datalength from the first byte of the first macroblock and the last non-zero byte of the last macroblock in the slice. This datalength must exclude all the extra zero padding at the end of a slice bitstream.	
		Bits [31:24] must be programmed to 0.	
2	31:29	Reserved	
		Format: MBZ	
	28:0	Indirect Data Start Address This field specifies the Graphics Memory starting address of the data to be fetched into BSD for processing. This pointer is relative to the BSD Indirect Object Base Address. Hardware ign this field if indirect data is not present. It is a byte-aligned address for the MPEG2 VLD bitstr data This address points to the first byte of the MB layer data, i.e. not including slice header.	D Unit nores ream
34	31:0	Inline Data All the required Slice Header parameters and error handling settings are captured as MPEG2_BSD_OBJECT Inline Data Descriptor structures. It has a fixed size of 2 DWs. Its definit described in the next section.	ition is



MFD_VC1_BSD_OBJECT

Source: VideoCS

2

Length Bias:

The MFD_VC1_BSD_OBJECT command is the only primitive command for the VC1 Decoding Pipeline. The macroblock data portion of the bitstream is loaded as indirect data object.Before issuing a MFD_VC1_BSD_OBJECT command, all VC1 states of the MFD Engine need to be valid. Therefore the commands used to set these states need to have been issued prior to the issue of a MFD_VC1_BSD_OBJECT command. VC1 deblock filter kernel cross the slice boundary if in the last MB row of a slice, so need to know the last MB row of a slice to disable the edge mask. There is why VC1 BSD hardware need to know the end of MB address for the current slice. As such no more phantom slice is needed for VC1, as long as the driver will program both start MB address in the current slice and the start MB address of the next slice. As a result, we can also support multiple picture state commands in between slices.

DWord	Bit			Descript	ion			
0	31:29	Command Type						
		Default Value: 3h PARALLEL_V			IDEO_PII	PE		
		Format:	ОрСо	ode				
	28:27	Pipeline						
		Default Value:		2h MFX_M	MULTI_DW			
		Format:		OpCode				
	26:24	Media Command Opcode						
		Default Value:				_DEC		
		Format:				e		
	23:21	SubOpcode A						
		Default Value:				1h		
		Format:				OpCode		
	20:16	SubOpcode B						
		Default Value:				8h		
		Format:				OpCode		
	15:12	Reserved						
		Format:		MBZ				
	11:0	DWord Length						
		Default Value:	0003h	Excludes D	DWord (0,1)			
		Format:	=n Tot	al Length -	1 - 2			
1	31:24	Reserved						
		Format:			1	VIBZ		
	23:0	Indirect BSD Data Length						
		Format:				U24		
		This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Data Start Address field is ignored. This to must have the same alignment as the Indirect Object Data Start Address. Long Format: It is the length in bytes of the bitstream data for the current slice/nicture. It includes the first byte of the						



				MFD_VC1_BSD_OBJE	СТ							
		first mace zero-pad contents Level bits length in Emulatio contents	irst macroblock and the last byte of the last macroblock in the slice/picture. Specifically, the ero-padding bytes (if present) and the next start-code are excluded. Hardware ignores the ontents after the last non-zero byte (trailing zeros). This field is sized to support VC1 AP@L4 evel bitstream. It includes the byte that contains the First MB Bit Offset Short Format: It is the ength in bytes of the bitstream data for the current slice, including Picture/Slice Header + imulation Prevention Bytes + any filling trailing zeros after the last MB. Hardware ignores the contents after the last non-zero byte. Trailing zero is allowed and handled correctly.									
2	31:29	Reserved	d		1							
		Format:			MBZ							
	28:0	Indirect	Data Sta	rt Address								
		Format:		GraphicsAddress[28:0]								
		This field Unit for ignores bitstrear	d specifie processir this field m data.	s the Graphics Memory starting address ig. This pointer is relative to the MFD Inc if indirect data is not present. It is a byte	of the data to be fetched into direct Object Base Address. Ha -aligned address for the VC1	o BSD ardware						
				Value	Name							
		[0,512M	B)									
3	31:24	Reserved	d									
		Format:			MBZ							
	23:16	Slice Sta This field macroblo Slice Laye	rt Vertica specifies ocks. For S er. This fie	al Position the position in y-direction of the first m SecondField this value is reset to zero as eld is for both Long and Short VC1 Interf	acroblock in the Slice in unit oppoed to the VC1 spec Ref: face Format.	of 9.1.2						
	15:9	Reserved	d									
		Format:			MBZ							
	8:0	Next Slice Vertical Position This field specifies the position in y-direction of the first macroblock in the next Slice in unit of macroblocks. This field is primarily used for error concealment. In the case that current slice is talast slice, this field should set to the height of picture (since y-direction is zero-based numbering)This field is maintained and provided by the driver for both Long and Short VC1 Interface Format.										
4	31:16	First_MB For DXVA bitstream	B_Byte_O A2 VC1 SI In for a slid	Ifset of Slice Data or Slice Header nort Format onlyIt gives the byte offset t e, relative to the Indirect BSD Data Start	o locate the first MB data in t Address.	he						
	15:5	Reserved	d									
		Format:			MBZ							
	4	Emulatio	on Preve	ntion Byte Present								
		Value	Name	Description	1							
		0h		H/W needs to perform Emulation Byte	Removal							



		MFD_VC1_BSD_OBJE	СТ				
	1h	H/W does not need to perform Emulation	on Byte	Removal			
3	Reserved						
	Format:		MBZ				
2:0	FirstMbBitOffse	t (First Macroblock Bit Offset)					
	Format:			U3			
	This field provide compressed bitst	s the bit offset of the first macroblock of ream. It is used with First_MB_Byte_Offset	the Slice for non	in the first byte of the input -byte aligned position.			



MFD_VC1_LONG_PIC_STATE

Source: VideoCS

Length Bias:

MFX_VC1_LONG PIC_STATE command encapsulates the decoding parameters that are read or derived from bitstream syntax elements above (inclusive) picture header layer. These parameters are static for a picture and when slice structure is present, these parameters are not changed from slice to slice of the same picture. Hence, this command is only issued at the beginning of processing a new picture and prior to the VC1_*_OBJECT command. The values set for these state variables are retained internally across slices. Only the parameters needed by hardware (BSD unit) to decode bit sequence for the macroblocks in a picture layer or a slice layer are presented in this command. Other parameters such as the ones used for inverse transform or motion compensation are provided in MFX_VC1_PRED_PIPE_STATE command. Driver will need to perform addition operations to generate all the fields in this command.

DWord	Bit				Descripti	ion				
0	31:29	Command Typ	е							
		Default Value: 3h PARALLE			3h PARALLEL_VI	_VIDEO_PIPE				
		Format:			OpCode					
	28:27	Pipeline								
		Default Value:		2h MFD_VC1_LONG_PIC_			ATE			
		Format:		OpCode						
	26:24	Media Comma	nd Opcode							
		Default Value:	Default Value:			2h VC1	_DEC			
		Format: Or				OpCod	e			
	23:21	SubOpcode A								
		Default Value:					1h			
		Format:					OpCode			
	20:16	SubOpcode B								
		Default Value:					1h			
		Format:					OpCode			
	15:12	Reserved								
		Format: MBZ								
	11:0	DWord Length								
		Default Value:			0004h Excludes D	Excludes DWord (0,1)				
		Format:			=n Total Length -	th - 2				
1	31:24	Reserved								
		Format:				1	MBZ			
	23:16	PictureHeightI	nMBsMinus1 (Pic	ture Height Minu	us 1 in M	Macroblocks)			
		Format:					U8			
		This field indica	tes the height c	of th	ne picture in unit c	of macro	bblocks. For example, for a 1920x1080			
		frame picture, P specified as 108	lictureHeightink 38 instead). This	vi⊔s fie	s equals 68 (1080 o Id is used in VLD a	divided and IT m	by 16, and rounded up, i.e. effectively nodes.			
		Value	Name			De	escription			



MFD_VC1_LONG_PIC_STATE

		[0,255]		a valid range of [0,255] [1, 256] M	В			
				Programming	Notes				
		Note: Even the	ough the Adva	anced Profile allows fran	ne dimensio	ons (width, height) to not be			
		use 'intermedia	ate buffer' tha	t is macroblock aligned	for decodi	na. In order to simplify the out-			
		of-bound refe	ence pixel acc	cess, the out-of-bound e	extrapolatio	on rule in VC1 spec can be used			
		to expand the	expected deco	oded frame to the inter	mediate bu	ffer dimension.			
	15:8	Reserved							
		Format:			MBZ				
	7:0	PictureWidthI	nMBsMinus1	(Picture Width Minus	1 in Macro	blocks)			
		Format:			U8-1				
		This field indica	tes the width	of the picture in unit of	macrobloc	ks. For example, for a 1920x1080			
		frame picture, F	PictureWidthIn	MBs equals 120 (1920 o	divided by 1	L6). This field is used in VLD and			
		11 modes		Nama		Description			
			e	Name		Description			
		[0,255]			[1,230] IVID				
2	31:24	Bitplane Buffe	r Pitch Minus						
		Format: U/-1 Pitch in (Bytes - 1). Specifies the bitplane buffer pitch in (#Dites - 1). Ditplane buffer is a linear buffer it is used at							
		only when the bitplane is not encoded as raw, and therefore is present in the header explicitly. In							
		VC1 Long Format, it is written by an application and later read by the HW. But in VC1 Short							
		Format, it is written and read by H/W only. This field is specified for better performance							
		Value Name							
		[0,FFFFFFFF]							
		Programming Notes							
		Programming Notes							
		The pitch must be equal to PictureWidthInMBs/2.For VC1 Long Format: The pitch must be equal to PictureWidthInMBs/2 For VC1 Short Format: If Pic Width is less than or equal to 2K							
		pixels, bitplane pitch is set to 64 (one cacheline; programmed as 63) bytes per MB row. If Pic							
		Width is greater than 2K pixels, bitplane pitch is set to 128 (two cachelines; programmed as							
		127) bytes per	MB row. This	field is not used in IT m	ode, used i	n VLD mode only. For VC1			
		DXVA2 Short Format, the bitplane specification is between H/W and Driver only. For Long							
	22.10								
	23:16	Reserved							
	4.5				IVIDZ				
	15	DmvSurfaceVa	the DMV read	d surface is valid. This s	urface store	d the direct motion vectors and			
		Mb type. This fi	eld is set for E	pictures that can refer	to a previo	us P picture for DMV. If there is			
		an I-picture bef	ore a B (in de	coding order) then this	field is not	set (as a result, zero's DMV's will			
		be assumed wh	ile decoding t	he B picture. That is, the	ere is no ex	plicit DMV buffer for an I-			
		picture). Whne	the current pi	cture being decoded is	an I, P or BI	, this bit is set to 0, since there is			
		no Diviv read in	i these picture	e decoding process. This	stiela is not	used in 11 mode, used in VLD			



MFD_VC1_LONG_PIC_STATE

14 ImplicitQuantizer Derived by driver from QUANTIZER. This field is used in this implementation's VC1 VLD Lo Format only, not used in IT and DXVA2 VC1.This bit is set to 1 when syntax element QUANTIZER=0, else its set to 0 13 Interpolation Rounder Contro Used only in MC operation. This field specifies the rounding control value used in interpol operation of motion prediction process. This field is used in VLD and IT modes. Programming Notes This bit field is taken from bRcontrol in DXVA_PictureParameters data structure 12 SyncMarker Indicates whether sync markers are enabled/disabled. If enable, sync markers "may be" prithe current video sequence being decoded. It is a sequence level syntax element and is varied for Simple and Main Profiles. Value Name Description 0h Not Present Sync Marker is not present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream 1h Maybe present Sync Marker and based in DVA2 VCI Main Profile, SyncMarker must set to 0.For Main Profile, SyncMarker can be set to 0 or 1.This field is used in both this implementation's an DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field incitates one of the following mo		mode on	ly.						
13 Interpolation Rounder Contro Used only in MC operation. This field specifies the rounding control value used in interpol operation of motion prediction process. This field is used in VLD and IT modes. Programming Notes This bit field is taken from bRcontrol in DXVA_PictureParameters data structure 12 SyncMarker Indicates whether sync markers are enabled/disabled. If enable, sync markers "may be" pritte current video sequence being decoded. It is a sequence level syntax element and is va for Simple and Main Profiles. Value Name Description 0h Not Present Sync Marker is not present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream This field is only valid in VLD mode. For Simple Profile, SyncMarker must set to 0.For Maii Profile, SyncMarker can be set to 0 or 1.This field is used in both this implementation's ar DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter-pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VC1 Spec. Value Name Description 0XX0b Chro	14	ImplicitQuantizer Derived by driver from QUANTIZER. This field is used in this implementation's VC1 VLD Long Format only, not used in IT and DXVA2 VC1. This bit is set to 1 when syntax element QUANTIZER=0, else its set to 0							
Programming Notes This bit field is taken from bRcontrol in DXVA_PictureParameters data structure SyncMarker Indicates whether sync markers are enabled/disabled. If enable, sync markers "may be" price urrent video sequence being decoded. It is a sequence level syntax element and is va for Simple and Main Profiles. Value Name Description 0h Not Present Sync Marker is not present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream This field is only valid in VLD mode. For Simple Profile, SyncMarker must set to 0.For Mair Profile, SyncMarker can be set to 0 or 1.This field is used in both this implementation's an DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter -pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VC1 Spec. Value Name Description 0XX0b Chroma Half-pel + Luma bicubic. (can only be 1MV) 0XX0b Chroma Half-pel + Luma bilinear Programming Notes <t< td=""><th>13</th><td>Interpola Used only operation</td><td>ation Roι y in MC o η of motic</td><td>Inder Cont peration. T</td><td>tro This field specifies the rounding control value used in inter on process. This field is used in VLD and IT modes.</td><td>polation</td></t<>	13	Interpola Used only operation	ation Roι y in MC o η of motic	Inder Cont peration. T	tro This field specifies the rounding control value used in inter on process. This field is used in VLD and IT modes.	polation			
This bit field is taken from bRcontrol in DXVA_PictureParameters data structure 12 SyncMarker Indicates whether sync markers are enabled/disabled. If enable, sync markers "may be" price urrent video sequence being decoded. It is a sequence level syntax element and is vare for Simple and Main Profiles. Value Name Description 0h Not Present Sync Marker is not present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream This field is only valid in VLD mode. For Simple Profile, SyncMarker must set to 0.For Mair Profile, SyncMarker can be set to 0 or 1.This field is used in both this implementation's an DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter-pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VC1 Spec. Value Name Description 0XX0b Chroma Half-pel + Luma bicubic. (can only be 1MV) 0XX0b Chroma Half-pel + Luma bilinear. Value Name Description 0XX0b Chroma Half-pel + Luma bilinear.					Programming Notes				
12 SyncMarker Indicates whether sync markers are enabled/disabled. If enable, sync markers "may be" prite current video sequence being decoded. It is a sequence level syntax element and is valifor Simple and Main Profiles. Value Name Description 0h Not Present Sync Marker is not present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream 1h Maybe present Sync Marker can be set to 0 or 1.This field is used in both this implementation's an DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter-pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VCI Spec. Value Name Description 0XX0b Chroma Quarter -pel + Luma bilinear. (can only be 1MV) 0XX0b Chroma Half-pel + Luma bilin		This bit f	field is tak	en from bl	Rcontrol in DXVA_PictureParameters data structure				
Value Name Description 0h Not Present Sync Marker is not present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream Programming Notes This field is only valid in VLD mode. For Simple Profile, SyncMarker must set to 0.For Mair Profile, SyncMarker can be set to 0 or 1.This field is used in both this implementation's an DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter-pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VC1 Spec. Value Name Description 0XX0b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 0XX1b Chroma Half-pel + Luma bilinear. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bilinear Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters de structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N	12	SyncMar Indicates the current for Simple	ker whether nt video s e and Ma	sync marke sequence b in Profiles.	ers are enabled/disabled. If enable, sync markers "may be' being decoded. It is a sequence level syntax element and is	present in valid only			
Oh Not Present Sync Marker is not present in the bitstream 1h Maybe present Sync Marker maybe present in the bitstream Programming Notes This field is only valid in VLD mode. For Simple Profile, SyncMarker must set to 0.For Mair Profile, SyncMarker can be set to 0 or 1.This field is used in both this implementation's an DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter-pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VC1 Spec. Value Name Description 0XX0b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 0XX1b Chroma Half-pel + Luma bicubic. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bilinear VX1b Chroma Half-pel + Luma bilinear 1X18 Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters da structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chr motion. This field is used in both VLD and IT modes. 7 Ran		Value	Na	ame	Description				
In Maybe present Sync Marker maybe present in the bitstream Programming Notes This field is only valid in VLD mode. For Simple Profile, SyncMarker must set to 0.For Mair Profile, SyncMarker can be set to 0 or 1.This field is used in both this implementation's an DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter-pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VC1 Spec. Value Name Description 0XX0b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 0XX0b Chroma Half-pel + Luma bicubic. (can only be 1MV) 0XX1b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 1XX1b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bicubic. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bicubic. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bicubic. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bicubic. (as only be 1MV) 1XX1b		0h	Not Pres	ent	Sync Marker is not present in the bitstream				
Programming Notes This field is only valid in VLD mode. For Simple Profile, SyncMarker must set to 0.For Mair Profile, SyncMarker can be set to 0 or 1.This field is used in both this implementation's an DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter-pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VC1 Spec. Value Name Description 0XX0b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 0XX1b 0XX0b Chroma Half-pel + Luma bilinear. (can only be 1MV) 1XX1b 1XX1b Chroma Half-pel + Luma bilinear. (can only be 1MV) 1XX1b 1XX1b Chroma Half-pel + Luma bilinear Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters da structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic M 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chr motion. This field is used in both VLD and IT modes. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaler on-the-fly, if RangeReduct		1h	Maybe p	oresent	Sync Marker maybe present in the bitstream				
This field is only valid in VLD mode. For Simple Profile, SyncMarker must set to 0.For Mair Profile, SyncMarker can be set to 0 or 1.This field is used in both this implementation's an DXVA2 VLD interface, but not used in IT mode. 11:8 Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter-pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VC1 Spec. Value Name Description 0XX0b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 0XX1b 0XX1b Chroma Quarter -pel + Luma bilinear. (can only be 1MV) 1XX0b 1XX1b Chroma Half-pel + Luma bilinear. Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters da structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chromator. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on -the-fly, if RangeReduction is Enabled.					Programming Notes				
Motion Vector Mode This field indicates one of the following motion compensation interpolation modes for P a pictures. The MC interpolation modes apply to prediction values of luminance blocks and always in quarter-sample. For chrominance blocks, it always performs bilinear interpolatio either half-pel or quarter-pel precision. Before the polarity of Chroma Half-pel or Q-pel is reversed from DXVA2 Spec, now I have fixed it to match with DXVA2 VC1 Spec. Value Name Description 0XX0b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 0XX1b 0XX1b Chroma Half-pel + Luma bicubic. (can only be 1MV) 0XX1b 1XX0b Chroma Quarter -pel + Luma bilinear. (can only be 1MV) 1XX1b 1XX1b Chroma Half-pel + Luma bilinear Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters de structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chromatoin. This field is used in both VLD and IT modes. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on -the-fly, if RangeReduction is Enabled.		This field Profile, S DXVA2 \	d is only v SyncMarko /LD interf	alid in VLD er can be se ace, but nc	o mode. For Simple Profile, SyncMarker must set to 0.For Net to 0 or 1.This field is used in both this implementation's ot used in IT mode.	lain and MS			
Value Name Description 0XX0b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 0XX1b Chroma Half-pel + Luma bicubic. (can be 1MV or 4MV) 1XX0b Chroma Quarter -pel + Luma bilinear. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bilinear. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bilinear Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters date structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chroma motion. This field is used in both VLD and IT modes. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on-the-fly, if RangeReduction is Enabled.	11:8	Motion V This field pictures. always in either hal reversed	/ector M indicates The MC in quarter-s lf-pel or c from DX\	ode one of the nterpolatio sample. For juarter-pel (A2 Spec, n	e following motion compensation interpolation modes for on modes apply to prediction values of luminance blocks a r chrominance blocks, it always performs bilinear interpola precision. Before the polarity of Chroma Half-pel or Q-pe now I have fixed it to match with DXVA2 VC1 Spec.	P and B nd are ation with I is			
0XX0b Chroma Quarter -pel + Luma bicubic. (can only be 1MV) 0XX1b Chroma Half-pel + Luma bicubic. (can be 1MV or 4MV) 1XX0b Chroma Quarter -pel + Luma bilinear. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bilinear Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters da structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chroma motion. This field is used in both VLD and IT modes. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on-the-fly, if RangeReduction is Enabled.		Value	Name		Description	-			
0XX1b Chroma Half-pel + Luma bicubic. (can be 1MV or 4MV) 1XX0b Chroma Quarter -pel + Luma bilinear. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bilinear Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters da structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chroma motion. This field is used in both VLD and IT modes. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on-the-fly, if RangeReduction is Enabled.		0XX0b		Chroma Q	Quarter -pel + Luma bicubic. (can only be 1MV)	-			
1XX0b Chroma Quarter -pel + Luma bilinear. (can only be 1MV) 1XX1b Chroma Half-pel + Luma bilinear Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters da structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chroma to the field is used in both VLD and IT modes. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on-the-fly, if RangeReduction is Enabled.		0XX1b		Chroma H	lalf-pel + Luma bicubic. (can be 1MV or 4MV)	-			
IXX1b Chroma Half-pel + Luma bilinear Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters da structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chroma motion. This field is used in both VLD and IT modes. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on-the-fly, if RangeReduction is Enabled.		1XX0b		Chroma Q	Quarter -pel + Luma bilinear. (can only be 1MV)	-			
Programming Notes Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters dates structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chroma motion. This field is used in both VLD and IT modes. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on-the-fly, if RangeReduction is Enabled.		1XX1b		Chroma H	lalf-pel + Luma bilinear				
Bits 11:8 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters data structure. Bit 11 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic N 8 of Motion Vector Mode = 1 for half-sample Chroma motion = 0 for quarter-sample Chroma motion. This field is used in both VLD and IT modes. 7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on-the-fly, if RangeReduction is Enabled.					Programming Notes				
7 RangeReductionScale This field specifies whether the reference picture pixel values should be scaled up or scaled on-the-fly, if RangeReduction is Enabled.		Bits 11:8 structure 8 of Mot motion.	are taker e. Bit 11 o tion Vecto This field	n from bM\ f Motion V or Mode = is used in b	VprecisionAndChromaRelation in DXVA_PictureParameter 'ector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicub 1 for half-sample Chroma motion = 0 for quarter-sample both VLD and IT modes.	s data ic MCBit Chroma			
This field specifies whether the reference picture pixel values should be scaled up or scaled on-the-fly, if RangeReduction is Enabled.	7	RangeRe	ductionS	Scale					
		This field on-the-fl	specifies y, if Rang	whether th eReduction	ne reference picture pixel values should be scaled up or sc n is Enabled.	aled down			
Value Name Description		Value	Name	e	Description				



MFD_VC1_LONG_PIC_STATE

	0h	Sca	e down i	reference picture by factor of 2				
	1h	Sca	e up refe	erence picture by factor of 2				
				Programming Notes				
	This bit is c field is used and RANG the current is the same represents keep Rang picture coh	derived by driv d in both VLD EREDFRM synt t picture. RANC e as (bPicDeblo the state of th eReductionSca nerent.	er for Ma and IT m ax eleme GERED is ocked >> e forward le, Range	ain Profile only. Ignored in Simple and Advanced odes. This is derived by driver from the history or onts (i.e. of forward/preceding reference picture) a the same as (bPicOverflowBlocks >> 3) & 1. RAN 5) & 1. For the current picture is a B picture, this d/preceding reference picture onlyDriver is respon eReduction Enable and RANGERED Present Flag of	Profiles. This f RANGERED and those of IGEREDFRM s field onsible to of current			
6	RangeRedu	uction Enable						
	This field sp preceding (scale up or Element (D)	pecifies whethe or forward) ref down should b XVA_PicturePa	r on-the erence p oe perfor rameters	-fly pixel value range reduction should be perform icture. Along with RangeReductionScale to speci- med. It is not the same value as RANGEREDFRM bPicDeblocked bit 5) in the Picture Header.	med for the fy whether Syntax			
	Value	Name		Description				
	0h	Disable	Range r	eduction is not performed				
	1h Enable Range reduction is performed							
	Programming Notes							
	This field is Advanced history of F picture) an & 1. RANG picture, thi responsible Flag of cur	s for Main Prof Profile. This fie RANGERED and d those of the EREDFRM is the s field represent to keep Rang rent picture co	ile only. S Id is used RANGE current p le same a nts the st leReduct herent.	Simple Profile is always disable, and not applicab d in both VLD and IT modes. This is derived by dr REDFRM syntax elements (i.e. of forward/precedi picture. RANGERED is the same as (bPicOverflow as (bPicDeblocked >> 5) & 1.For the current picture tate of the forward/preceding reference picture o ionScale, RangeReduction Enable and RANGERED	le to river from the ing reference Blocks >> 3) ure is a B onlyDriver is D Present			
5	LOOPFILTE	R Enable Flag						
	This filed is Deblocking and also the MFX_PIPE_N operation for set to 0, but indicates th	the decoded s is ON accordin e loop filter un MODE_SELECT ollows the VC1 t PostDeblock(ne loop filter un	yntax ele ng to pic it.When t comman standard DutEnabl nit is used	ement LOOPFILTER in bitstream. It indicates if In- ture level bitstream syntax control. This bit affect this bit is set to 1, PostDeblockOutEnable field in id must also be set to 1. In this case, in-loop debl d - deblocking doesn't cross slice boundary. Whe e field in MFX_PIPE_MODE_SELECT command is s d for out-of-loop deblocking. In this case, debloc	loop s BSD unit locking en this bit is set to 1. It king			
	Value	Nar	ne	Description	noue.			
	Oh	Disable		Disables loop filter				
	1h	Enable		Enables loop filter				



				/IFD_	VC1_LONG_PIC_STATE					
		ON at t	he pictu	e levelThi	s field is used in both VLD and IT modes.					
		Valu	e	Name	Description					
		0h	Dis	able	to disable overlap smoothing filter					
		1h	Ena	ble	to enable overlap smoothing filter					
	3	Second This flag	field g is set fo	or the sec	ond field in field pictures. This field is used in both VLD and IT modes					
	2:1	Reserve	ed							
		Format	•		MBZ					
	0	VC1 Profile specifies the bitstream profile. This field is used in both VLD and IT modes.								
		Value	Name		Description					
		0h	Disable	current p Simple a	oicture is in Simple or Main Profile (No need to distinguish nd Main Profile)					
		1h	Enable	current p	victure is in Advanced Profile					
					Programming Notes					
		This is profile	required and alsc	because to find o	128 is added for intra blocks post inverse transform in advanced ut if Motion vectors are adjusted or not.					
3	31	Reserved								
		Format			MBZ					
	30:29	CondOv This field controls level qu VLD mo	d is the of the over the over alization	decoded s rlap smoo step size not in DX	syntax element CONDOVER in a bitstream of advanced profile. It othing filter operation for an I frame or an BI frame when the picture PQUANT is 8 or lower. This field is used in this implementation's VC1 (VA2 VC1 and IT modes.	L				
		Value	Name		Description					
		00b		No overla	ap smoothing					
		01b		Reserved						
		10b		Always p	erform overlap smoothing filter					
		11b		Overlap s	moothing on a per macroblock basis based on OVERFLAGS					
	11b Overlap smoothing on a per macroblock basis based on OVERFLAGS 28:26 PicType (Picture Type) This field specifies the coding type of the picture according to the Frame Coding Mo FCM = 00 01 (a Progressive or Interlaced Frame Picture):000 = I001 = P010 = B011 SkippedOther encodings are reservedWhen FCM = 10 11 (a Field Picture)000 = I/I0 = P/I011 = P/P100 = B/B101 = B/B1110 = BI/B111 = BI/BIAlthough, for a field picture a field-pair, but HW will only look at one field state only, and the other field state is of This field is read and qualified with the SecondField flag internally. This field is uniqui implementation's VC1 VLD Long format, and is used in IT mode as well. For DXVA2 \driver needs to convert the DXVA2 interface to this implementation's HW VLD Long interface.									
	25:24	FCM (Fi This is t	rame Co he same	ding Mo as the va	de) riable FCM defined in VC1.This field must be set to 0 for Simple and					



			MFD_	VC1_LONG_PIC_S	ΤΑΤΕ				
		Main Profile mode as we implementa	ain ProfilesThis field is unique to this implementation's VC1 VLD Long format, and is used in IT ode as well. For DXVA2 VC1 IT mode, driver needs to convert the DXVA2 interface to this uplementation's HW VLD Long Format interface.						
		Value	Name	Descriptio	on				
		00b	Disable	Progressive Frame Picture					
		01b	Enable	Interlaced Frame Picture					
		10b		Field Picture with Top Field First					
		11b Field Picture with Bottom Field First							
	23:21	Reserved Format: MBZ							
	20:16	AltPQuant This field is configuratic VOPDQUAN mode only,	tPQuant (Alternative Picture Quantization Value) is field is identical to the variable ALTPQUANT which is derived from VOPDQUANT nfiguration in the VC1 standard. This field must be set to 0 for Simple/Main I and BI pictures as DPDQUANT is not present. This field is used in this implementation's VC1 VLD Long Format ode only, not used in DXVA2 VC1 VLD and IT modes.						
	15:13	Reserved			r	1			
		Format:			MBZ				
	12:8	PQuant (Picture Quantization Value) Format: U5 This is the same as the calculated variable PQUANT in VC1 standard where PQuant = PQINDEX,							
		except when QUANTIZER = 0 and PQINDEX > 8, it is given asPQuant = (PQINDEX < 29) ? PQINDEX - 3: PQINDEX*2 - 31This field is used in all picture types (I, P, B and BI) and all operating modes (IT mode and this implementation's and DXVA2 VLD modes).							
	7:0	BScaleFactor BScaleFactorThis field is the scale factor for computing Direct-mode motion vectors. It is derived from the variable BFRACTION in the VC1 standard, section 8.4.5.4. There are only 21 valid values corresponding to the 21 encodings of BFRACTION as shown in the table here. Other values are reserved. MSB of this field can be used to determine if BFRACTION is greater than or equal to 1/2, which is used to determine Motion Prediction Type for B pictures. Effectively, condition "BFRACTION >= 1/2" is equivalent to condition "BScaleFactor >= 128". This field is only valid for B pictures. This field is used only in this implementation's VC1 VLD Long format mode, it is not used in DXVA2 VC1 VLD and IT modes. BFRACTION VLCBFRACTIONBScaleFactor0001/21280011/3850102/31700111/4641003/41921011/5511102/51021110003/515311100014/520411100101/64311100115/621511101001/73711101012/77411101103/71111101114/714811110005/718511110016/722211110101/83211110113/							
4	31:30	Reserved							
		Format:			MBZ				
	29:28	UnifiedMv This field is parsing Lun Mode). This	Mode (Unifie a combination na MVD from s field is also u	ed Motion Vector Mode) n of the variables MVMODE and N the bitstream. This field is used to used to signal Q-pel or Half-pel M	MVMODE2 in the VC1 sta o signal 1MV vs 4MVallow IVD read from the bitstrea	ndard, for ved (Mixed am. The			



		Μ	FD_VC	1_LC	NG_PIC_STATE				
	bicubic or bil Mode field, a mode only, it	bicubic or bilinear Luma MC interpolation mode is duplicate information from Motion Vector Mode field, and is ignored here. This field is used in this implementation's VC1 VLD Long Forma mode only, it is not used in DXVA2 VC1 VLD and IT modes.							
	Value		Name		Description				
	00b			Mixed M	IV, Q-pel bicubic				
	01b			1-MV, Q-pel bicubic 1-MV half-pel bicubic					
	10b								
	11b			1-MV ha	If-pel bilinear				
27	FourMvSwitch (Four Motion Vector Switch) This field indicates if 4-MV is present for an interlaced frame P picture. It is identical to the variable 4MVSWITCH (4 Motion Vector Switch) in VC1 standard. This field is used in this implementation's VC1 VLD Long Format mode only, it is not used in DXVA2 VC1 VLD and IT modes.					dentical to the used in this VC1 VLD and IT			
	Value		Nan	ne	Description				
	0h	h Disable			only 1-MV				
	1h		Enable		1, 2, or 4 MVs				
	It is derived f should have t Value Oh	rom F. the sai Nar	ASTUVMC me value a ne no re	= (bPicSp s Motion	oatialResid8 >> 4) & 1 in both VLD an Vector Mode LSBit. Description	d IT modes, and			
	1h		quar	ter-pel c	ffsets to half/full pel positions				
25	RefFieldPicPolarity (Reference Field Picture Polarity) This field specifies the polarity of the one reference field picture used for a field P picture. derived from the variable REFFIELD defined in VC1 standard and is only valid when one fie referenced (NUMREF = 0) for a field P picture. When NUMREF = 0 and REFFIELD = 0, this field the polarity of the reference I/P field that is temporally closest; When NUMREF = 0 and RE = 1, this field is the polarity of the reference I/P field that is the second most temporally cl The distance is measured based on display order but ignoring the repeated field if presen- to RFF = 1). This field is unique to this implementation's VC1 VLD Long format mode, and								
	Value		Name		Description				
	0h			Тор	(even) field				
	1h			Bott	om (odd) field				
24	NumRef (Nu This field indi identical to th (FCM = 10 1 not used in I	imber icates ne vari L1). Th Γ and I	of Refere how many iable NUMI is field is u DXVA2 VC1	nces) referenc REF in th nique to L modes.	e fields are referenced by the current e VC1 standard. This field is only valid this implementation's VC1 VLD Long	(field) picture. It is for field P picture format mode, and is			
	Value		Nan	ne	Description				



MFD	_VC1_	LONG	_PIC_STAT	Έ
-----	-------	------	-----------	---

	0h		One field	reference	ed		
	1h Two fields referenced						
23:20	BwdRefDist (Ref This field is valid o This field is uniqu and DXVA2 VC1 n	erence Distance only in B field pi e to this implen nodes.	:e) ictures giving th nentation's VC1	e value o VLD Long	f BRFD. The field is ignor g format mode, and is no	ed in P Picture. ot used in IT	
19:16	FwdRefDist (Reference Distance)						
	Format:				U4		
	This field is the number of frames between the current frame and its reference frame. It is derive from the syntax element REFDIST (P Reference Distance) in the VC1 standard. 0 means that the previous frame is the reference frame. It has the same value as of FRFD for both P and B field pictures. This field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.					ne. It is derived eans that the and B field and is not	
		Value			Name		
	[0, 15]						
15:12	Reserved	Reserved					
	Format:				MBZ		
	This field specifies the variable DMV VLD Long format	s the differentia RANGE in the V mode, and is no	I motion vector C1 standard. Th ot used in IT and	range in is field is d DXVA2	interlaced pictures. It is e unique to this implemen VC1 modes.	equivalent to Itation's VC1	
	Value	Name			Description		
	00b		No extended r	ange			
	01b		Extended horiz	ontally			
	10b		Extended verti	cally			
	11b		Extended in bo	oth directi	ons		
9:8	ExtendedMVRange (Extended Motion Vector Range Flag) This field specifies the motion vector range in quarter-pel or half-pel modes. It is equivalent to the variable MVRANGE in the VC1 standard. This field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes						
	Value	Name			Description		
	00b		[-256, 255] x [-:	.28, 127]			
	01b		512, 511] x [-25	6, 255]			
	10b		[-2048, 2047] x	[-1024, 1	023]		
	11b		[-4096, 4095] x	[-2048, 2	047]		
7:4	AltPQuantEdgeN This field is a bit r used for the edge DQUANTFRM, DQ as shown in Error! 0: Left picture edg	lask (Alternati nask for the fou macroblocks. I PROFILE, DQSE Reference sou ge macroblocks	ive Picture Qua ur edges in clock t is derived base BEDGE, DQDBEE rce not foundT Bit 1: Top pictur	ntization -wise orced on the OGE, and I his field is e edge m	Edge Mask) ler, indicating whether A following variables DQU DQBILEVEL defined in the s valid only if AltPQuant acroblocksBit 2: Right pi	ltPQuant is ANT, VC1 standard Config is 01. Bit cture edge	



			Μ	FD	_VC1_	LONG_	PIC_STATE		
		macroblo VC1 VLD	ocksBit 3: Long for	Botto mat n	om picture node, and i	edge macrob is not used in	locksThis field is unique to IT and DXVA2 VC1 modes.	this implementation's	
	3:2	AltPQuantConfig (Alternative Picture Quantization Configuration) This field specifies the way AltPQuant is used in the picture. It determines how to compute the macroblock quantizer step size, MQUANT. It is derived based on the following variables DQUANT, DQUANTFRM, DQPROFILE, DQSBEDGE, DQDBEDGE, and DQBILEVEL defined in the VC1 standard, as shown in Error! Reference source not foundThis field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes. Value Name Description 00b AltPQuant not used						now to compute the ving variables EVEL defined in the unique to this XVA2 VC1 modes.	
		01b		AltPO		ed and applie	ed to edge macroblocks on	<u>у</u>	
		100 11b		AltPO	Quant and	POuant are s	elected on macroblock basi	<u>s</u>	
	1	HalfQP This field is used for inverse quantization of AC coefficients. It is valid only when PQuant is used This field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.					when PQuant is used. Ind is not used in IT		
	0	Indicatin AC coeffi PQuantU cases: 1) and PQua and 2) QI mode, ar	PQuantUniform Indicating if uniform quantization applies to the picture. It is used for inverse quantization of the AC coefficients. QUANTIZER 001123PQUANTIZER01PQINDEX>=9<=8 PQuantUniform010201ImplicitQuantizer = 0, and PQuantUniform = 0 is used to represent 2 cases: 1) QUANTIZER=01 and PQUANTIZER=0; and 2) QUANTIZER = 10b. ImplicitQuantizer = 0, and PQuantUniform = 1 is used to represent 2 cases: 1) QUANTIZER=01 and PQUANTIZER=1; and 2) QUANTIZER = 11bThis field is unique to this implementation's VC1 VLD Long format						
		Va	lue		Name		Description		
		0h				Non-un	iform		
		1h				Uniform			
5	31	Bitplane This field fields list VC1_BSD are applie unique to VC1 mod	PresentF indicates ed in bits _BUF_BA cable for this imp les.	Elag (I s whe 22:10 SE_ST the co oleme	Bitplane Bo ther the bit 5 is coded i ATE comm urrent pictu ntation's V	uffer Presen plane buffer n non-raw m and points to ure in bits 22: C1 VLD Long	t Flag) is present for the picture. If ode, and Bitplane Buffer Ba o the bitplane buffer. Otherv 16 must be coded in raw ma format mode, and is not us	set, at least one of the se Address field in the vise, all the fields that ode. This field is red in IT and DXVA2	
		Valu	ue	Na	ame		Description		
		0h			b	itplane buffe	r is not present		
		lh			b	itplane buffe	r is present		
	30	Forward This field only valid format m	MbRaw indicates d when Pi node, and	s whe icture is no	ther the FC Type is B. T t used in IT	RWARDMB f his field is ur and DXVA2	field is coded in raw or non- nique to this implementatior VC1 modes.	raw mode. This field is n's VC1 VLD Long	
			Value		Na	ame	Descript	tion	



MFD_VC1_LONG_PIC_STATE	MFD_	VC1_	LONG	PIC	STATE
------------------------	------	------	------	------------	-------

	Т						
	0h		non-raw mode				
	1h		raw mode				
29	9 MvTypeMbRaw This field indicates whether the MVTYPREMB field is coded in raw or non-raw mode. only valid when PictureType is P. This field is unique to this implementation's VC1 VLI format mode, and is not used in IT and DXVA2 VC1 modes.						
	Value	Name	Description				
	0h		Non-Raw Mode				
	1h		Raw Mode				
28	SkipMbRaw This field indicates wh valid when PictureTyp implementation's VC1	iether the SKIPMB field is be is P or B.0 = non-raw i L VLD Long format mode	s coded in raw or non-raw mode. This field is only mode1 = raw modeThis field is unique to this e, and is not used in IT and DXVA2 VC1 modes.				
	Value	Name	Description				
	0h	Disable	Non-Raw Mode				
	1h	Enable	Raw Mode				
	only valid when Pictur format mode, and is r Value	reType is P or B. This field not used in IT and DXVA2 Name	d is unique to this implementation's VC1 VLD Long 2 VC1 modes. Description				
	0h		Non-Raw Mode				
	1h		Raw Mode				
26	OverflagsRaw This field indicates wh only valid when Pictur format mode, and is r	ether the OVERFLAGS fin reType is I or BI. This field not used in IT and DXVA2	eld is coded in raw or non-raw mode. This field is d is unique to this implementation's VC1 VLD Long 2 VC1 modes.				
	Value	Name	Description				
	0h		Non-Raw Mode				
	1h		Raw Mode				
25	AcPredRaw This field indicates wh valid when PictureTyp format mode, and is r	AcPredRaw This field indicates whether the ACPRED field is coded in raw or non-raw mode. This field is only valid when PictureType is I or BI. This field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.					
	Value	Name	Description				
	0h	Disable	Non-Raw Mode				
	1h	Enable	Raw Mode				
24	FieldTxRaw This field indicates wh valid when PictureTyp	iether the FIELDTX field i be is I or BI. This field is u	s coded in raw or non-raw mode. This field is only nique to this implementation's VC1 VLD Long				



MFD VC1 LONG PIC STATE Value Name Description 0h Disable Non-Raw Mode 1h Enable Raw Mode 23 Reserved Format: MBZ 22:20 MvTab (Motion Vector Table) U3 Format: This field specifies which motion vector table(s) is (are) used for motion vector (differential) decoding in a P or B picture. This field is the combination of the variables MVTAB and IMVTAB in the VC1 standard. Two bits are defined for progressive frame pictures; And two or three bits are defined for interlaced field/frame pictures depending on NUMREF and P or B picture types. This field is valid for P and B pictures. It is not valid for I pictures. For P or B progressive frame pictures0 = Motion Vector Differential VLD Table 01 = Motion Vector Differential VLD Table 12 = Motion Vector Differential VLD Table 23 = Motion Vector Differential VLD Table 3The other encodings are reservedFor P interlace field pictures with NUMREF = 0 or P/B interlace frame pictures0 = 1-Reference Table 01 = 1-Reference Table 12 = 1-Reference Table 23 = 1-Reference Table 3The other encodings are reservedFor P interlace field picture with NUMREF = 1 or Binterlaced field pictures 0 = 2-Reference Table 01 = 2-Reference Table 12 = 2-Reference Table 23= 2-Reference Table 34 = 2-Reference Table 45 = 2-Reference Table 56 = 2-Reference Table 67 = 2-Reference Table 7The other encodings are reservedThis field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes. 19:18 **FourMvBpTab (4-MV Block Pattern Table)** This field specifies which table is used to decode the 4-MV block pattern (4MVBP) syntax element in 4-MV macroblocks. It is identical to the variables 4MVBPTAB in the VC1 standard, section 9.1.1.37. This field is valid only in interlace frame P, B pictures, or interlace field P, B pictures. It is not valid for I picture. For interlace field P and B pictures, it is only valid if UnifiedMvMode is equal to Mixed-MV Type. For interlace frame P picture, it is only valid if FourMvSwitch is 1.For interlace frame B picture, it is always valid.0 = 4MVBP Table 01 = 4MVBPTable 12 = 4MVBP Table 23 = 4MVBP Table 3This field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes. 17:16 **TwoMvBpTab (2MV Block Pattern Table)** This field specifies which table is used to decode the 2MV block pattern (2MVBP) syntax element in 2MV field macroblocks. It is identical to the variables 2MVBPTAB in the VC1 standard, section 9.1.1.36. This field is valid only in interlace frame P/B pictures. It is not valid for I picture, nor for interlace field P or B pictures.0 = 2MVBP Table 01 = 2MVBP Table 12 = 2MVBP Table 23 = 2MVBP Table 3This field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes. 15:14 **Reserved** Format: MBZ 13:12 **TransType (Picture-level Transform Type)** U2 Format: This field specifies the Transform Type at picture level. It is identical to the variable TTFRM in the

VC1 standard, section 7.1.1.41. This field is only valid when TransTypeMbFlag is 1. Otherwise, it is



		MFD_VC	1_LONG_PIC_STATE					
	reserved and 8x8 Transform this implemer	reserved and MBZ. This field is set to 00 when VSTRANSFORM is 0 in the entry point layer.00 = 8x8 Transform01 = 8x4 Transform10 = 4x8 Transform11 = 4x4 TransformThis field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.						
11	TransTypeMI This field india level. It is ider when VSTRAN VC1 VLD Long	TransTypeMbFlag (Macroblock Transform Type Flag) This field indicates whether Transform Type is fixed at picture level or variable at macroblock level. It is identical to the variable TTMBF in the VC1 standard, section 7.1.1.40.This field is set to 1 when VSTRANSFORM is 0 in the entry point layer. This field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.						
	Value	Name	Description					
	0h	varia	ble transform type in macroblock layer					
	1h	use p	picture level transform type TransType					
10:8	MbModeTab (Macroblock Mode Table) This field signals which code table is used to decode the macroblock mode syntax element (MBMODE) in the macroblock layer in a P or B picture. This field is identical to the variables MBMODETAB in the VC1 standard, section 9.1.1.33. This field is valid for interlace frame P, B picture and interlace field P, B picture. It is not valid for I picture, nor progressive frame P, B pictures. Two bits are defined for interlace frame P, B pictures; And three bits are defined for interlaced field P, B pictures. Two bits are defined for interlace frame P, B pictures. There are two set of code tables selected based on if UnifiedMvMode is equal to 4-MV Type or not. 0 = Code Table 01 = Code Table 12 = Code Table 23 = Code Table 3Other encodings are invalidThree bit are defined for interlace field P, B pictures. There are two set of code tables selected based on i UnifiedMvMode is equal to Mixed-MV Type or not. 0 = Code Table 01 = Code Table 12 = Code Table 23 = Code Table 34 = Code Table 45 = Code Table 56 = Code Table 67 = Code Table 7TH field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DY/A2 VC1 media.							
7:6	TransAcY (Pi BitFieldDesc	cture-level Tran	sform Luma AC Coding Set Index, TRANSACTA	ABLE2				
5:4	TransAcUV (I This field, togo the non-zero variables TRAI same as TRAN be programm set index 01 = implementatio	itFieldDesc TransAcUV (Picture-level Transform Chroma AC Coding Set Index, TRANSACTABLE) his field, together with PQINDEX, specifies which intra AC coding set to be used for decoding he non-zero AC coefficients in a coded luma (Y) block. This field is the combination of the ariables TRANSACFRM and TRANSACFRM2 in the VC1 standard. For I pictures, TransAcY is the ame as TRANSACFRM2. For other pictures, it is the same as TRANSACFRM, and therefore must he programmed to be the same as TransAcUV. This field is valid for all picture types.0 = Coding et index 01 = Coding set index 12 = Coding set index 23 is invalidThis field is unique to this mplementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 medee						
3	TransDcTab (This field spec the Transform TRANSDCTAB is unique to th VC1 modes. Value Oh	TransDcTab (Intra Transform DC Table) This field specifies whether the low motion tables or the high motion tables are used to decode the Transform DC coefficients in intra-coded blocks. This field is identical to the variable TRANSDCTAB in the VC1 standard, section 8.1.1.2.This field is valid for all picture types. This field is unique to this implementation's VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes. Value Name Description 0h The high motion tables						
	1h		The low motion tables					

Г



2:0 CbpTab (Coded Block Pattern Table) This field specifies the table used to decode the CBPCY syntax element for each coded		MFD_VC1_LONG_PIC_STATE
macroblock in P and B pictures. This field is combination of the variable CBPTAB for P and B frame pictures and the variable ICBPTAB in interlace field P, B pictures and interlace frame P, B pictures in the VC1 standard (Table 52 and Table 102). This field is reserved and MBZ for I or BI pictures as I only has a fixed table.000 = Table 0 (Table 169 for P, B frames or Table 124 otherwise)001 = Table 1 (Table 170 for P, B frames or Table 125 otherwise)010 = Table 2 (Table 171 for P, B frames or Table 126 otherwise)011 = Table 3 (Table 172 for P, B frames or Table 127 otherwise)100 = Table 4 (Table 128 for interlace field/frame P, B pictures)101 = Table 5 (Table 129 for interlace field/frame P, B pictures)110 = Table 6 (Table 130 for interlace field/frame P, B pictures)111 = Table 7 (Table 131 for interlace field/frame P, B pictures)This field is unique to this involvement time b VC1 by D and for mercine and is not used by VCA2 VC1 mercine	2:0	CbpTab (Coded Block Pattern Table) This field specifies the table used to decode the CBPCY syntax element for each coded macroblock in P and B pictures. This field is combination of the variable CBPTAB for P and B frame pictures and the variable ICBPTAB in interlace field P, B pictures and interlace frame P, B pictures in the VC1 standard (Table 52 and Table 102). This field is reserved and MBZ for I or BI pictures as I only has a fixed table.000 = Table 0 (Table 169 for P, B frames or Table 124 otherwise)001 = Table 1 (Table 170 for P, B frames or Table 125 otherwise)010 = Table 2 (Table 171 for P, B frames or Table 126 otherwise)011 = Table 3 (Table 172 for P, B frames or Table 127 otherwise)100 = Table 4 (Table 128 for interlace field/frame P, B pictures)101 = Table 5 (Table 129 for interlace field/frame P, B pictures)110 = Table 7 (Table 131 for interlace field/frame P, B pictures)This field is unique to this implementation b/(A2)/(A2)/(A2)/(A2)/(A2)/(A2)/(A2)/(A2)



		MFD_	/C1_	SHORT_PI		TATE		
Source:		VideoCS						
Length E	Bias:	2						
DWord	Bit			Descripti	ion			
0	31:29	Command Type						
		Default Value:		3h PARALLEL_VI	DEO_P	PIPE		
		Format:		OpCode				
	28:27	Pipeline						
		Default Value:	2ł	n MFD_VC1_SHORT	_PIC_S	TATE		
		Format:	Format: OpCode					
	26:24	Media Command Opco	de					
		Default Value:			2h VC	1_DEC		
		Format:			OpCo	de		
	23:21	SubOpcode A						
		Default Value:				1h		
		Format:				OpCode		
	20:16	SubOpcode B						
		Default Value:				0h		
		Format:				OpCode		
	15:12	Reserved						
	11:0	DWord Length		0002h Eveludes D		(0.1)		
		Default value.	=n Total Length - 2					
1	21.24							
T	31:24	Format:				MBZ		
	22.10					IND2		
	23:16	Format:	Picture	Height in Macrobic	ocks			
		This field indicates the he	eight of t	the picture in unit c	of maci	roblocks. For example, for a 1920x1080		
		frame picture, PictureHeightInMBs equals 68 (1080 divided by 16, and rounded up, i.e. effectively						
		specified as 1088 instead). This field is used in VLD and IT modes. Note: Even though the						
		Advanced Profile allows frame dimensions (width, height) to not be aligned to macroblock						
		that is macroblock aligne	ed for de	coding. In order to	simpli	fy the out-of-bound reference pixel		
		access, the out-of-bound	l extrapc	lation rule in VC1	spec ca	in be used to expand the expected		
		decoded frame to the int	ermedia	te buffer dimension	n. 	Description		
		10 2551		Name	[1 25	61 MB		
	15.0	Pocorriad			[1, 23			
	72;8	Format:				MBZ		
		- Stringt.						



		MFD_V	C1_SHORT_PI	C_STATE						
	7:0	Picture Width								
		Format: U8-1 Picture Width in Macroblocks								
	macroblocks. For example, for a 1920x1080									
		frame picture, PictureWidthInMBs equals 120 (1920 divided by 16). This field is used in VLD								
		Value	Namo	Description						
		I0 2551	Name	I1 2561 MB						
2	21.74	Ritplana Ruffar Bitch Min	uc 1							
2	51.24	Format:	U7-1 Pitch in Bytes							
		Specifies the bitplane buffe	er pitch in (#Bytes - 1). Bit	plane buffer is a linear buffer. It is needed						
		only when the bitplane is n	ot encoded as raw, and the	nerefore is present in the header explicitly. In						
		VC1 Long Format, it is writt	en by an application and	later read by the HW. In VC1 Long Format, it						
		read by H/W only. This field	d is specified for better pe	rformance. For VC1 Long Format: The pitch						
		must be equal to PictureW	idthInMBs/2. For VC1 Sho	rt Format: If Pic Width is less than or equal						
		to 2K pixels, bitplane pitch	is set to 64 (one cacheline	; programmed as 63) bytes per MB row. If						
		Pic Width is greater than 2 127) bytes per MB row. Thi	s field is not used in IT mo	et to 128 (two cachelines; programmed as						
		Short Format, the bitplane	specification is between F	I/W and Driver only. For Long Format,						
		application is responsible f	or allocation with the driv	er.						
	23	Interpolation Rounder Control								
	23	Used only in MC operation. This field specifies the rounding control value used in interpolation								
		operation of motion prediction process.								
		This field is used in VLD and IT modes.								
	22:20	Reserved								
		Format:	MBZ							
	19:16	Motion Vector Mode								
		This field indicates one of the following motion compensation interpolation modes for P and B								
		pictures. The MC interpolation modes apply to prediction values of luminance blocks and are								
		always in quarter-sample. For chrominance blocks, it always performs bilinear interpolation with either half-pel or quarter-pel precision 0XX0 = Chroma Quarter -pel + Luma bicubic (cap only be								
		1MV)0XX1 = Chroma Half-	pel + Luma bicubic. (can l	pe 1MV or 4MV)1XX0 = Chroma Quarter -pel						
		+ Luma bilinear. (can only l	pe 1MV)1XX1 = Chroma H	lalf-pel + Luma bilinearNote: Bits 19:16 are						
		taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters data structure. Bit 19								
		Vector Mode = 1 for half-s	ample Chroma motion =	0 for guarter-sample Chroma motion. This						
		field is used in both VLD ar	nd IT modes.	· ·						
	15	DmvSurfaceValid								
		Indicated when the DMV re	ead surface is valid. This su	urface stored the direct motion vectors.						
		before a B (in decoding or	der) then this field is not s	et (as a result, zero's DMV's will be assumed						
		while decoding the B pictu	re. That is, there is no exp	licit DMV buffer for an I-picture).						
		This field is not used in IT	mode, used in VLD mode	only.						



	MFD_VC1_SHORT_PIC_STATE							
14:12	Reserved Format:			MBZ				
11	VC1 Profile specifies the b Note: This is profile and als This field is u	C1 Profile Decifies the bitstream profile. Note: This is required because 128 is added for intra blocks post inverse transform in advanced rofile and also to find out if Motion vectors are adjusted or not. This field is used in both VLD and IT modes.						
	Value Na	ne	on					
	0h [Def a	current pic ult] Simple and	ture is in Simple or Main Pro I Main Profile)	ofile (No need to distinguish				
	1h	current pic	ture is in Advanced Profile					
10:6	Reserved			1	1			
	Format: MBZ							
5	Backward Provide A still need to p This field is u bPicBackward The Intra Pict picture type, a IT mode.	ackward Prediction Present Flag ote: a B picture that only uses forward prediction may have this flag set to 1 as well. Driver may ill need to provide a valid reference picture index. his field is used in both DXVA2 VC1 VLD mode and IT mode. It is the same parameter as PicBackwardPrediction in DXVA2 VC1 spec. The Intra Picture Flag, Backward Prediction Present Flag and RefPicFlag are used to derive the ficture type, as specified in PTYPE for a frame, and in FPTYPE for a field, in DXVA2 VC1 VLD and						
4	Intra Picture This field is us bPicIntra in D The Intra Pict picture type, a IT mode.	ntra Picture Flag his field is used in both DXVA2 VC1 VLD mode and IT mode. It is the same parameter as PicIntra in DXVA2 VC1 spec. The Intra Picture Flag, Backward Prediction Present Flag and RefPicFlag are used to derive the picture type, as specified in PTYPE for a frame, and in FPTYPE for a field, in DXVA2 VC1 VLD ar						
	Value Name	e	Description					
	0h	entire picture o MB type.	ntire picture can have a mixture of intra and inter MB type or just inter B type.					
	1h	entire picture i	s coded in intra MB type					
3	SecondField This flag is set	t for the second f	ïeld in field pictures. This fie	eld is used in both VLD and IT r	nodes.			
2	Reserved			1				
	Format:			MBZ				
1:0	Picture Struc This field is us bPicStructure The Picture S specified in F0	ture sed in both DXVA in DXVA2 VC1 sp tructure and Prog CM, in DXVA2 VC	A2 VC1 VLD mode and IT mo bec. gressive Pic Type are used to C1 VLD and IT mode.	ode. It is the same parameter as o derive the picture structure a	s			
	Value	Name		Description				
	01b		top field (bit 0)					



MFD_VC1_SHORT_PIC_STATE							
		10b		bc	ottom field (bit 1)	om field (bit 1)	
	11b			fra	ame (both fields are	ne (both fields are present)	
	00b			ille	egal		
3	31 Reserved						
		Format: MBZ					
	30	Overlap Smoothing Enable Flag This field is the decoded syntax element OVERLAP in bitstreamIndicates if Overlap smoothing is ON at the picture levelThis field is used in both VLD and IT modes					
	Value Name De				Desci	cription	
	0h Disable to disable overlap sn				le overlap smoothir	lap smoothing filter	
		1h	Enable	to enab	le overlap smoothin	overlap smoothing filter	
	29	Range Reduction Scale					
		Access:				None	
		down on-the-fly, if RangeReduction is Enabled. NOTE: This bit is derived by driver for Main Profile only. Ignored in Simple and Advanced Profiles. This field is used in both VLD and IT modes. This is derived by driver from the history of RANGERED and RANGEREDFRM syntax elements (i.e. of forward/preceding reference picture) and those of the current picture. RANGERED is the same as (bPicOverflowBlocks >> 3) & 1. RANGEREDFRM is the same as (bPicDeblocked >> 5) & 1. For the current picture is a B picture, this field represents the state of the forward/preceding reference picture onlyDriver is responsible to keep RangeReductionScale, RangeReduction Enable and RANGERED Present Flag of current picture coherent.					
	Value Name			j	Description ale down reference picture by factor of 2		
	0h Disable [Default] Sc		cale down reference				
		1h	Enable	nable Sca		le up reference picture by factor of 2	
	28	Range Reduction EnableThis field specifies whether on-the-fly pixel value range reduction should be performed for the preceding (or forward) reference picture. Along with RangeReductionScale to specify whether scale up or down should be performed. It is not the same value as RANGEREDFRM Syntax Element (DXVA_PictureParameters bPicDeblocked bit 5) in the Picture Header. This field is for Main Profile only. Simple Profile is always disable, and not applicable to Advanced Profile. This field is used in both VLD and IT modes. This is derived by driver from the history of RANGERED and RANGEREDFRM syntax elements (i.e. of forward/preceding reference picture) and those of the current picture. RANGERED is the same as (bPicOverflowBlocks >> 3) & 1. RANGEREDFRM is the same as (bPicDeblocked >> 5) & 1.For the current picture is a B picture, this field represents the state of the forward/preceding reference picture onlyDriver is responsible to keep RangeReductionScale, RangeReduction Enable and RANGERED Present Flag of current picture coherent.ValueNameDescription					
		Oh	Disable [Defa	lt]	Range reduction	is not performed	
	1h Enable Range reduction is performed				is performed		
	27:24	:24 Reserved					


MFD_VC1_SHORT_PIC_STATE

	Format	t:		MBZ							
23:22	Progres This fiel bPicExtr	ressive Pic Type field is used in both DXVA2 VC1 VLD mode and IT mode. It is the same parameter as extrapolation in DXVA2 VC1 spec. The Picture Structure and Progressive Pic Type are used to									
Value Name Description											
	0			proares	ssive only picture						
	1			progres	ssive only picture						
	2		i	interlace picture (frame-interlace or field-interlace)							
	3		i	illegal							
21	Reserve	ed	1								
	Format	t:			MBZ						
20:16	P-Pic R	ef Dist	ance								
	Access				None						
	level fla BI/B, BI, 0.This fi modes.	= 1, and if the picture type is not one of the following types: B/B, B/BI, I flag REFDIST_FLAG == 0, REFDIST shall be set to the default value of 2 VC1 VLD mode only, not used in IT and our VC1 VLD Long Format									
	value				Name						
	0-10										
4 - 4 4											
15:14	QUAN				Description						
	00h	Name	impli	icit qua	antizer at frame leve						
	01b		expli non-	cit qua uniforn	antizer at frame level, and use PQUANTIZER SE to specify uniform or m						
	10b		expli	cit qua	antizer, and non-uniform quantizer for all frames						
	11b		expli	cit qua	ntizer, and uniform quantizer for all frames						
13	MULTI	RES Pr	esent	Flag (fo	or Simple/Main Profile only)						
	Valu	e l	lame		Description						
	0h RESPIC Parameter is present in the picture header										
	1h			RESPIC Parameter is present in the picture header							
12	SYNCM	IARKE	R Pres	ent Fla	ag (for Simple/Main Profile only)						
	Value	Na	me		Description						
	0		В	litstrear	m for Simple and Main Profile has no sync marker						
	1		В	Bitstream for Simple and Main Profile may have sync marker(s)							
11	RANGERED Present Flag (for Simple/Main Profile only)										



		N	/IFD_V	C1_SHORT_PIC_STATE				
	It is nee RangeR	ader Parsing. Driver is responsible to keep RangeReductionScale, ad RANGERED Present Flag of current picture coherent.						
	Value	lue Name Description						
	0		Range Red header	uction Parameter (RANGEREDFRM) is not present in the picture				
	1		Range Red	uction Parameter (RANGEREDFRM) is present in the picture header.				
10:	8 MAXBF Number	RAMES	secutive B I	rames.				
7	PANSC	AN Pres	sent Flag					
	Value	Nar	ne	Description				
	0		Pan So	can Parameters are not present in the picture header				
	1		Pan So	can Parameters are present in the picture header				
6	REFDIS For head VC1 VLE	F_FLAG der pars) mode:	sing REFDIS s.	T. This is used in DXVA2 VC1 VLD mode only, not used in IT and our				
	Deblock and also MFX_PIF operatic set to 0, indicate operatic	ing is C the loc PE_MOE on follov but Pos s the loo on does	ON accordin op filter uni DE_SELECT of ws the VC1 stDeblockC op filter un cross slice	ig to picture level bitstream syntax control. This bit affects BSD unit t.When this bit is set to 1, PostDeblockOutEnable field in command must also be set to 1. In this case, in-loop deblocking standard - deblocking doesn't cross slice boundary. When this bit is putEnable field in MFX_PIPE_MODE_SELECT command is set to 1. It it is used for out-of-loop deblocking. In this case, deblocking boundary. This field is used in VLD mode only, not in IT mode.				
	Valu	ie	Name	Description				
	0	0		In-Loop-Deblocking-Filter is disabled				
	1			In-Loop-Deblocking-Filter is enabled				
4	4 FastUVMCFlag (Fast UV Motion Compensation Flag) This field specifies whether the motion vectors for UV is rounded to half or full p identical to the variable FASTUVMC in VC1 standard. This field is used in both VL It is derived from FASTUVMC = (bPicSpatialResid8 >> 4) & 1 in both VLD and IT should have the same value as Motion Vector Mode LSBit.							
	Value	e Na	ame	Description				
	0h		no r	ounding				
	1h		quarter-pel offsets to half/full pel positions					
3	EXTEND BitField	Desc	/ Present F	lag				
	Value	Description						
	0h		Extend	ded_MV is not present in the picture header				
	1h		Extend	ded_MV is present in the picture header				
2:1		DQUANT						



MFD_VC1_SHORT_PIC_STATE

		Access	:		None					
		Format	t:		U2					
		Use for	Picture Hea	ader P	Parsing of VOPDUANT elements					
		Value	Name		Description					
		0h	[Default]							
		00b		no V quan	no VOPDQUANT elements; Quantizer cannot vary in frame, same quantization step size POUANT is used for all MBs in the frame					
		01b		refer	to VC1 Spec. for all the MB position dependent quantizer selection					
		10b		The r with PQU	macroblocks located on the picture edge boundary shall be quantized ALTPQUANT while the rest of the macroblocks shall be quantized with ANT.					
		11b	Reserved							
	0	VSTRA	NSFORM f	ad						
	Ū	Valu	e Nar	ne	Description					
		0h	Disable	variable-sized transform coding is not enabled						
		1h	Enable		variable-sized transform coding is enabled					
4	31:29	Reserve	ed							
		Format	t: MBZ	(for p	ossible future change to BFraction Enumeration)					
	28:24	BFraction Enumeration This field is the scale factor for computing Direct-mode motion vectors. It is derived from the variable BFRACTION in the VC1 standard, section 8.4.5.4. There are only 21 valid values corresponding to the 21 encodings of BFRACTION as shown in the table here. Other values are reserved. The VLD decoded value of BFRACTION (from the picture header) is mapped into an enum value from 0 to 20.(??? MSB of this field can be used to determine if BFRACTION is greater than or equal to 1/2, which is used to determine Motion Prediction Type for B pictures. Effectively, condition "BFRACTION >= 1/2" is equivalent to condition "ScaleFactor >= 128". ??? How can the enum replicate this feature ???)This field is only valid for B pictures. This field is used only in DXVA2 VC1 VLD mode, it is not used in our VC1 VLD Long Format mode and IT mode. BFRACTION VLCBFRACTION = 1/2"11110102/7121110103/571110014/ 5811100101/6911100115/61011101001/7111110112/71211101103/71311101114/ 71411110005/71511110016/7161111010/81711110113/81811111005/81911111017/ 8201111111BL Pic Indicator31 (optional)								
	23	Reserved								
	22.22	Format		vance	a rione only, KANGE_MAPT_FLAG Kange Mapping not supported					
	22:20	Format	ed MR7 Ac	lvance	ed Profile only: RANGE MAPY Range Manning not supported					
	10	Pess		vance						
	19	Format	ed t: MBZ Adv	/ance	d Profile only; RANGE_MAPUV_FLAG Range Mapping not supported					
	18:16	Reserve	eserved							



		Μ	FD_V	/C1_SHORT_PIC_STATE					
	Format: MBZ Advanced Profile only; RANGE_MAPUV Range Mapping not supported								
15:9	Reserve	d							
	Format:			MBZ					
8	4MV All	owed F	lag						
7	POSTPR	OC Flag	I						
6	PULLDC	WN							
5	INTERL	ACE							
4	TFCNTR	FLAG							
3	FINTER	LAG							
	For a BI picture, REFPIC flag must set to 0For I and P picture, REFPIC flag must set to 0.For a B picture, REFPIC flag must set to 0, except for a B-field in interlaced field mode which can be 0 or 1 (e.g. the top B field can be used as a reference for decoding its corresponding bottom B-field in a field pair). In VLD mode, this flag cannot be used as an optimization signaling for an I or P picture that is not used as a reference picture. This field is used in both DXVA2 VC1 VLD mode and IT mode. It is the same parameter as bPicDeblockConfined[bit2] in DXVA2 VC1 spec.The Intra Picture Flag, Backward Prediction Present Flag and RefPicFlag are used to derive the picture type, as specified in PTYPE for a frame, and in FPTYPE for a field, in DXVA2 VC1 VLD and IT mode.								
	0h		the curre	ent picture after decoded, will never used as a reference picture					
	1h the current picture after decoded, will be used as a reference picture later								
1	PSF								
0	EXTENDED_DMV Present Flag								
	Value	N	ame	Description					
	0h	[Def	ault]	Extended_DMV is not present in the picture header					
	1h			Extended_DMV is present in the picture header					



		MFX_AVC_DI	RE	стмос)E_S	STATE			
Source: VideoCS									
Length E	Bias:	2							
This is a picture level command and is issued once per picture. All DMV buffers are treated surfaces, in which the lower 6 bits are used for conveying surface states. Current Pic POC be available in POCList[32 and 33] of the MFX_AVC_DIRECTMODE_STATE Command. This in the AVC decoding in VLD and IT modes, and AVC encoder mode. The same command and Short DXVA2 AVC Interface. The DMV buffers are not required to be programmed for					Fers are treated as standard media rrent Pic POC number is assumed to ommand. This command is only valid ne command supports both Long ogrammed for encoder mode.				
DWord	Bit	Description							
0	31:29	Command Type							
		Default Value:	3h P	ARALLEL_VIDE	EO_PIF	ΡĒ			
		Format:	ОрС	ode					
	28:27	Pipeline							
		Default Value:		2h MFX_SING	ile_d\	N			
		Format:		OpCode					
	26:24	Media Command Opcode							
		Default Value:				1h AVC			
		Format:				OpCode			
	23:21	L SubOpcodeA							
		Default Value:			0h M	EDIA_			
		Format:			ОрСо	ode			

	20:16	SubOpcodeB							
		Default Value:			2h Desc				
		Format:			OpCode				
	15:12	Reserved							
		Format:		MBZ					
	11:0	DWord Length	DWord Length						
		Default Value:		0043h Excludes DWord (0,1)					
		Format:		=n Total Length - 2					
1	31:6	Direct MV Buffer Base A	ddress	for Picture 0 (current o	r reference top field)				
		Format:	Graph	nicsAddress[31:6]					
		This field provides the base address of the DMV write buffer to store motion vectors decoded in the current picture (top field), which may be used later as a collocated motion information read buffer of the associated reference picture in decoding subsequent B-pictures that have MB coded in direct mode. It is a private buffer used by the MPR hardware only. Its content is not accessed by software. This buffer must be 64-byte cacheline aligned. The write buffer size is							

Г



		MFX_A	VC_DIRECTMC	DDE_STA	FE		
	the corresponding collocated DMV and motion information. For reference picture, these 32 DMV read Buffers can be indexed by the frame_store_ID[4:0], which is obtained from RefPicList L0/L1[RefPicIdx]. frame_Store_IDbit[0] (indicator for Top/Bottiom Field). For writing out motion information during the decoding of the current picture, all 34 DMV buffers can be addressed by [img_dec_fs_idc[4:0] <<1 + img_structure[1]].						
5:4	Direct M	/ Buffer - Arb	itration Priority Control	the GAC/GAM p	ipeline for this sur	face.	
	Valu	e	Name	<u> </u>	Descript	ion	
	00b	Highest	priority		Desc		
	01b	Second	highest priority		Desc		
	10b	Third hi	ghest priority				
	11b	Lowest	priority				
			Duo anno in	- Notes			
	This field	of Picture 0 D	Programmir MV Buffor must always be	ig Notes	ardlass if this huff	for is active	
	or not, ex 34 possib	ist or not. H/V De DMV buffe	V only reads this bit to deter rs. This field is ignored in al	ermine the arbitra I the other DMV	ation priority contr buffers 1 to 33.	ol for all	
3	Reserved						
2	Direct MV This field of GTT. The e is ignored	/ Buffer - Gra contains the G effective GFDT for reads.	phics Data Type (GFDT) f FDT bit for this surface whe is the logical OR of this fie	or Picture 0 en writes occur. G Id with the GFDT	FDT can also be so from the GTT entr	et by the y. This field	
	١	/alue	Name		Description		
	0h		Disable	Desc			
	1h		Enable				
			Drogrammin				
	This field	of Picture 0 D	MV Buffer must always be	programmed rec	ardless if this huff	fer is active	
	or not, ex buffers. T	kist or not. H/V his field is ign	V only reads this bit to dete ored in all the other DMV k	ermine the GFDT ouffers 1 to 33.	for all 34 possible	DMV	
1:0	Direct M This field	/ Buffer - Cac controls cache	heability Control for Pictor ability.	ure 0		1	
	Value		Name		Description		
	00bUse cacheability control bits from GTT entryDesc01bData is not cachedDesc						
	11b	Data is cached	b				
			Programmir	ig Notes			
	This field or not, ex possible	of Picture 0 D tist or not. H/V DMV buffers. 1	MV Buffer must always be V only reads this bit to dete Fhis field is ignored in all th	programmed, rec ermine the cachea e other DMV buf	gardless if this buff ability control for a fers 1 to 33.	fer is active all 34	



		MFX_AVC_DIRECTMODE_STATE						
2	31:6	Direct MV Buffer Base Address for Picture 1 (current or reference bottom field)						
		Format: GraphicsAddress[31:6]						
		This field provides the base address of the DMV read/write buffer for the current or reference						
		picture (bottom field). It is paired with the DMV Buffer of Picture 0 for MB pair retrieval during						
		read. It follows the same format specification as DMV buffer for Picture 0It is only valid if the						
		current picture is a bottom field. It is also valid						
	5:4	Direct MV Buffer - Arbitration Priority Con						
		Format: U2						
		This field is ignored in H/W, and assumes the same value as of Picture 0 DMV Buffer specification bit[5:4] above.						
	3	Reserved						
	2	Direct MV Buffer - Graphics Data Type (GFDT) for Picture 1						
		Format: U1						
		This field is ignored in H/W, and assumes the same value as of Picture 0 DMV Buffer specification						
		bit[2] above.						
	1:0	Direct MV Buffer -Cacheability Control for Picture 1						
		Format: U2						
		This field is ignored in H/W, and assumes the same value as of Picture 0 DMV Buffer specification bit[1:0] above.						
332	31:6	Direct MV Buffer Base Address for Reference Frame 2 to 31						
		Format: GraphicsAddress[31:6]						
		This field provides the base address of the DMV buffer for reference frame 2 to 31. They are						
		needed if the current B-Picture has MBs coded in direct mode. It is a private buffer used by the						
		MPR hardware only. Its content is not accessed by software. All these buffers must be 64-byte						
		cacheline aligned. There are a total of 32 possible Direct MV Read Buffers (not including the						
		size is 557.056 bytes for 1 frame (the selected colPic). Scalable with frame height, but do not						
		scale with frame width as the hardware assumes frame width (in MBs) fixed at 128 (smallest						
		power of 2 value larger than 120 - 1920x1088 screen resolution). The adjacent DMV buffers are						
		paired ([2 and 3], [4 and 5], [N and N+1],[30 and 31]).						
	5:4	Direct MV Buffer - Arbitration Priority Control						
		Format: U2						
		This field is ignored in H/W, and assumes the same value as of Picture 0 DMV Buffer specification						
		bit[5:4] above.						
	3	Reserved						
	2	Direct MV Buffer - Graphics Data Type (GFDT) for Reference Frame 2 to 31						
		Format:						



		MFX_AVC_DIRECTMODE_STATE								
		This field is ignored in H/W, and assumes the same value as of Picture 0 DMV Buffer specification bit[2] above.								
	1:0	Direct MV Buffer - Cacheability Control for Reference Frame 2 to 31								
		Format: U2								
		his field is ignored in H/W, and assumes the same value as of Picture 0 DMV Buffer specification it[1:0] above.								
3334	31:6	Direct MV Buffer Base Addresses 32 and 33 (Write-Only Buffer), for Current Decoding Frame/Field								
		Format: GraphicsAddress[31:6]								
		Inis field provides the base address of the DMV write-only buffer for the current decoding frame/field. It is a private buffer used by the MPR hardware only. Its content is not accessed by software. All these buffers must be 64-byte cacheline aligned, i.e. the same as the above DMV read/write buffers. These 2 buffers can only be addressed by [img_dec_fs_idc[4:0] < 1 + img_structure[1]] for the current picture being decoded. Each write buffer size is 557,056 bytes for 1 frame (the selected colPic). Scalable with frame height, but do not scale with frame width as the hardware assumes frame width (in MBs) fixed at 128 (smallest power of 2 value larger than 120 - 1920x1088 screen resolution). DMV write buffer 32 is valid only if the current picture is a progressive frame, MbAff frame, or a top field. DMV write buffer 33 is valid only if the current picture is a bottom field.								
	5:4	Direct MV Buffer 32 and 33 (Write-only Buffer) - Arbitration Priority Control								
		Format: U2								
		This field is ignored in H/W, and assumes the same value as of Picture 0 DMV Buffer specification bit[5:4] above.								
	3	Reserved This field is ignored for writes.								
	2	Direct MV Buffer 32 and 33 (Write-only Buffer) - Graphics Data Type (GFDT) for Current Frame/Field								
		Format: U1								
		This field is ignored in H/W, and assumes the same value as of Picture 0 DMV Buffer specification bit[2] above.								
	1:0	Direct MV Buffer 32 and 33 (Write-only Buffer) - Cacheability Control for Current Frame/Field								
	Format: U2									
		This field is ignored in H/W, and assumes the same value as of Picture 0 DMV Buffer specification bit[1:0] above.								
3568	31:0	POC List, POCList[34][31:0] Each POC value is a signed 32-bit number. One-to-one correspondence with the 34 Direct MV Buffer Address for Reference and Currrent Frames/FieldsThere are 34 POC entries in the list. For reference picture, only the lower 32 POC [0-31] entries can be used, and POCList[] is indexed by								



MFX_AVC_DIRECTMODE_STATE						
	the frame_store_ID[4:0], which is obtained from RefPicList L0/L1[RefPicIdx]. frame_Store_IDbit[0] (indicator for Top/Bottiom Field). For current picture, all 34 POC entries [0-33] can be addressed by POCList[img_dec_fs_idc[4:0] < 1 + img_structure[1]]. For frame-only mode, every other entry is skipped. For MBAFF and field-only picture, each entry is a field POC, and every two entries are paired.					

г



			MFX_AV	C_IMO	G_STATE			
Source:	Vio	deoCS						
Length Bias:	Bias: 2							
This must be the commands	very firs	st command	to issue after th	e surface :	state, the pipe so	elect and base address setting		
DWord	Bit				Description			
0	31:29	Command	Туре	I				
		Default Val	ue:	3h P	ARALLEL_VIDEO	_PIPE		
		Format:		OpCo	ode			
	28:27	Pipeline						
		Default Va	ue:	2h N	1FX_AVC_IMG_S	TATE		
		Format:		OpC	ode			
	26:24	Media Con	nmand Opcode		1			
		Default Val	ue:		1h AVC_COMM	10N		
		Format:			OpCode			
	23:21	SubOpcode	e A			1		
		Default Val	ue:			0h		
		Format:				OpCode		
	20:16	SubOpcode	e B			1		
		Default Val	ue:			0h		
		Format:				OpCode		
	15:12	Reserved						
		Format:				MBZ		
	11:0	DWord Ler	ngth OCh Euclusian Di	M =				
		Value:	UCh Excludes D	word (U,1)				
		Format:	=n 00Eh, used f	or normal	decode and end	code mode000h, a special case to		
			in DW1 which is	s part of th	e dummy image	e state command are ignored by		
		hardware.						
1	31:16	Reserved						
		Format: MBZ						
	15:0	Frame Size						
		Format: U16-1 in MB unit						
		The value for FrameSizeInMBs must match the product of FrameWidthInMBs						
		in MB unit	. This paramete	er is speci	fied for Intel in	nterface only.		
		Value	Name		D	escription		
		[0,16383]		represent	ing Number of	MBs [1,16384]		
2	31:24	Reserved	·					



MFX_AVC_IMG_STATE

		Format:			MBZ			
	23:16	Frame Height	Frame Height					
		Format:		U8-1 in MB unit				
		It is set to the value of (FrameHeightInMBsMinus1+ 1). Since the max value for FrameHeightInMBs is 255, the max allowed value for FrameHeightInMBsMinus1 i 254. The min value for FrameHeightInMBs is 1.Although the max. value that can be specified for FrameHeightInMBs is 255 (in the current implementation), FrameWidthInMBs * FrameHeightInMBs must not exceed the max value of FrameSizeInMBs[14:0].e.g. for 1920x1080, FrameHeightInMBs[7:0] is equal to 68 (divided by 16, and rounded up, i.e. effectively specified as 1088 instead). It is derif from FrameHeightInMbs = (2 - frame_mbs_only_flag) * PicHeightInMapUnits an PicHeightInMbs = FrameHeightInMbs / (1 + field_pic_flag) internally done. For N PicHeightInMapUnits is in MB pair unit, so the bitstream sends only half frame he						
		Value	Name		Description			
		[0,255]		representing height	[1,256]			
	15:8	Reserved						
		Format:			MBZ			
	7:0	Frame Width						
		Format:		U8-1 in MB unit				
		It is set to the va FrameWidthInMI 254. The min val specified for Fran FrameWidthInMI FrameSizeInMBs divided by 16, ar from FrameWidt PicWidthInMbs = PicWidthInMapL	lue of (FrameW Bs is 255, the m meWidthInMBs Bs * FrameWid [14:0].e.g. for 1 nd rounded up, hInMbs = (2 - = FrameWidthI Inits is in MB p	AdthInMBsMinus1+ 1 nax allowed value for idthInMBs is 1.Althou is 255 (in the current thInMBs must not exc 920x1080, FrameHeig i.e. effectively specifi frame_mbs_only_flag nMbs / (1 + field_pic_ air unit, so the bitstre). Since the max value for FrameWidthInMBsMinus1 is only gh the max. value that can be implementation), eed the max value of htInMBs[7:0] is equal to 68 (1080 ed as 1088 instead). It is derived) * PicWidthInMapUnits and _flag) internally done. For MBAFF, am sends only half frame width.			
		Value	Name		Description			
		[0,255]		representing width	[1,256]			
3	31:29	Reserved						
		Format:			MBZ			
	28:24	24 Second Chroma QP Offset Signed integer value. It should be in the range of -12 to +12 (according to AVC specifies the offset for determining QP Cr from QP Y. It is set to the upper 5 bits of value of the syntax element (Chroma_qp_offset[9:0]) read from the current active Chroma_qp_offset [4:0] - chroma_qp_offset_bits (from the current active PPS)Chroma_qp_offset [9:5] - second_chroma_qp_offset_bits						
	23:21	Reserved			1			
		Format:			MBZ			
	20:16	First Chroma QI Signed integer v	P Offset alue. It should	be in the range of -12	to +12 (according to AVC spec). It			



			MFX_A	AVC_IMG_STATE				
		specifie value of Chroma PPS)Chi	specifies the offset for determining QP Cb from QP Y. It is set to the lower 5 bits of the value of the syntax element (Chroma_qp_offset[9:0]) read from the current active PPS. Chroma_qp_offset [4:0] - chroma_qp_offset_bits (from the current active PPS) Chroma_qp_offset [9:5] - second chroma_qp_offset bits					
1	L5:14	Reserve	ed					
		Format	••	MBZ				
	13	Reserve	ed					
_		Format	••	MBZ				
	12	Weight	ed_Pred_Flag					
		Format	••	Enable				
		Value	Name	Description				
		0	Disable [Default]	specifies that weighted prediction is not used for P and SP slices				
		1	Enable	specifies that weighted prediction is used for P and SP slices				
		TI-1- C-		Programming Notes				
			and must set to	0 for B and I pictures.				
1	11:10	Weight	ed_BiPred_Idc	Description				
		0		Specifies that the default weighted prediction is used for B				
		0	[Default]	slices				
		1	EXPLICIT	Specifies that explicit weighted prediction is used for B slices				
		2	IMPLICIT	Specifies that implicit weighted prediction is used for B slices.				
		3	Reserved	Illegal value				
				Programming Notes				
_		This fie	ld must set to (0 for P and I pictures.				
	9:8	ImgStr The cur	uct - Image Str rent encoding p	ructure, img_structure[1:0] picture structure can only takes on 3 possible values				
			Value	Name				
		00b		Frame Picture				
		01b		I OP Field Picture				
		105		Invalid not allowed				
		100						
				Programming Notes				
		img_st	ructure[0] can b	be used as a flag to distinguish between frame and field				
		structu	structure. It must be consistent with the field_pic_flag setting in the Slice Header. This					



			MFX	X_A	VC_IMG_ST	ATE		
		parameter is specified for this interface only.						
	7:0	Reserved						
		Format:				MBZ		
		Used to b	e 8-bit i	img_d	lec_fs	I		
4	31:16	MinFram	eWSize					
		Default V	/alue:				0h	
		Format:						
	15	Minimu Frame Siz need to p CABAC_Z parameter represen FrameBi mode. The prog When Mi Program Program Program	m Fram ze is sp perform ZERO_V er, not ted by tRateN mable mable mable mable	ne Siz pecifie n emu VORE part o Minir Max (able r eWSiz range range	te [15:0] (in Word, 1 ed to compensate for ulation byte insertion) 0 insertion (if any) at t of DXVA. The caller sh num Frame Size, is alv DWORD 10 bits 29:1 ange 02^18-1 zeUnits is 00. e is 02^20-1 when N e is 02^26-1 when N e is 02^32-1 when N	6-bit) (Encoder (Rate Control. Cu is done only to the last slice of a ould always mak vays less than ma 6). This field is re AinFrameWSizeU AinFrameWSizeU	Dnly) Mininum rrently zero fill (no the end of the picture. Encoder e sure that the value, eximum frame size served in Decode nits is 01. nits is 10. nits is 11.	
		Format:	11			Enable		
		pass enco first pass t	ider, all to 0, it c	passe does s	s except the first one ne ave some memory band	eed to set this valu width.	e to 1. By setting the	
		Value	Nar	me		Description		
		0	Disabl	le	Disable Reading of Ma	croblock Status Bu	ffer	
		1	Enable	9	Enable Reading of Mac	roblock Status Buf	fer	
	14	LoadSlice	Pointe	rFlag		Γ		
		Format:				Enable		
		LoadBitStr additional set to 0, b subseque output da of a frame different r	oadBitStreamPointerPerSlice (Encoder-only)To support multiple slice picture and additional header/data insertion before and after an encoded slice. When this field is set to 0, bitstream pointer is only loaded once for the first slice of a frame. For subsequent slices in the frame, bitstream data are stitched together to form a single output data stream. When this field is set to 1, bitstream pointer is loaded for each slice of a frame. Basically bitstream data for different slices of a frame will be written to different memory locations.					
		Value N	lame			Description		
		0 D	isable I	Load I	BitStream Pointer only c	once for the first sli	ce of a frame	
		1 Er	nable I	Load/	reload BitStream Pointe	r only once for the	each slice, reload the	



		MF	X_AV	C_IMG_STATE		
			start location of the bitstream buffer from the Indirect PAK-BSE Obje Data Start Address field			
13	Reserved					
12	MvUnpa MVUnPac	kedFla kedEna	g ble (Encod	der Only)This field is reserved in Decoo	le mode.	
	Value	N	lame	Description		
	0	PACKE	D	use packed MV format (compliant to	DXVA)	
	1	UNPA	CKED	use unpacked 8MV/32MV format on	у	
11:10	ChromaF Chroma F compone	ormatI ormat I nt (Cb,)	dc DC, Chron Cr) in the	naFormatIdc[1:0]It specifies the sampli current picture as follows :	ng of chroma	
	Value			Name	Descript	ion
	00b	mo	nochrome	picture	Desc	
	01b	4:2:	0 picture		Desc	
	10b	4:2:	2 picture ((not supported)		
	11b	4:4:	4 picture ((not supported)		
				Programming Notes		
	It is set to	o the va	alue of the	syntax element read from the current	active SPS Th	ie
	correspo	nding N	Aonochroi	me Flag (monochrome_flag) can be de	rived from this	s field.
9	Reserved					
	Format:			MBZ		
8	MbMvFo	rmatFla	ag			
	Use MB le	evel Mv	Format fla I	g (Encoder Only)		1
	Value	Name		Description		-
		INORE	 DRE HW PAK ignore MvFormat in the MB data. When bit 12 == 0, all MBs use packed MV formatWhen 12 == 1, each MB data must use unpacked MV format, 8MV when there is no minor MV involved, and 32MV if 			
	1 F	OLLOW	HW PAK data.	will follow MvFormat value set within	each MB	
	Programming Notes					
	They mu and the 3 MvUnpa	hey must take one of the two values: the 8MV unpacked format (MvFormat =101b), nd the 32MV unpacked format (MvFormat =110b). This bit can be set only when AvUnpackedFlag (bit 12 of this register) is set otherwise system could hang				:101b), hen
7	EntropyC	odingF	lag			
	Entropy C	oding F	lag, entro	py_coding_flag	_	
	Value			Name	_	
	0	CAV	LC bit-seri	al encoding mode		



		MFX_	AVC	_IMG_ST	ATE				
	1 CABAC bit-serial encoding mode.								
	Programming Notes								
	It specifies one of the two possible bit stream encoding modes in the AVC. It is set to the value of the syntax element read from the current active PPS								
 6	ImgDis	posableFlag	hle Flac	or Non-Reference	ce Picture Flag				
	Value	Name			Description				
	0	REFERENCE	the cu for ot	rrent decoding p	icture may be used a	as a reference picture			
	1	DISPOSABLE	the cu (e.g. a subse	rrent decoding p B-picture cannot quent decoding)	icture is not used as be a reference pictu	a reference picture ure for any			
				Programmi	ng Notes				
	It is de elemer written	rived from Im nt from a NAL out.This field	ngDispo - unit. V d is only	sableFlag = (nal_ı /hen this flag is se valid for VLD dee	ref_idc == 0). nal_ref et, no reference pictu coding mode.	_idc is a syntax ure and DMV are			
5	Constra Constra syntax e	ainedIPredFl ined Intra Pre element in the	ag ediction e currer	Flag, constrained t active PPS.	l_ipred_flagIt is set to	o the value of the			
	Value	Name			Description				
	0 INTRA_AND_INTER allows both intra and inter neighboring MB to be used the intra-prediction encoding of the current MB.					ng MB to be used in current MB.			
	1	INTRA_ONLY	(allows only to us prediction encod an inter MB, it is	e neighboring Intra ling of the current M considered as not av	MBs in the intra- IB. If the neighbor is vailable.			
4	Direct8x8InfFlag Direct 8x8 Inference Flag, direct_8x8_inference_flagIt is set to the value of the syntax element in the current active SPS. It specifies the derivation process for luma motion vectors in the Direct MV coding modes (B_Skip, B_Direct_16x16 and B_Direct_8x8). When frame_mbs_only_flag is equal to 0, direct_8x8_inference_flag shall be equal to 1.It must be consistent with the frame_mbs_only_flag and transform_8x8_mode_flag								
	Value	Name			Description				
	0	SUBBLOCK	allows s 4x8)	ubpartitioning to	go below 8x8 block	size (i.e. 4x4, 8x4 or			
	1	BLOCK	allows p block si	processing only at ze.	8x8 block size. MB	Info is stored for 8x8			
3	DIOCK SIZE. Transform8x8Flag 8x8 IDCT Transform Mode Flag, trans8x8_mode_flagSpecifies 8x8 IDCT transform may be used in this pictureIt is set to the value of the syntax element in the current active PPS.								



			MF	X_A	AVC_IN	/IG_STA	TE		
		Value	Value Name Description						
		0	4x4	no 8x	וס 8x8 IDCT Transform, only 4x4 IDCT transform blocks are preser				
		1	1 8x8 8x8 Transform is allowed						
	2	FrameMbOnlyFlag Frame MB only flag, frame_mbs_only_flagIt is set to the value of the syntax elemer					t to the value of the syntax element in		
		Value	Name	e ses.		D	escription		
		0	FALSE I	not tru	ie ; effective	ly enables the	e possibility of MBAFF mode.		
		1	TRUE t	true, o MBAFI	true, only frame MBs can occur in this sequence, hence disallows the				
	1	MbaffF MBAFF (mb_ada syntax e current mbaff_fi all the s the fielc img_stru 0 1	baffFlameFlag IBAFF mode is active, mbaff_frame_flag. It is derived from MbaffFrameFlag = ib_adaptive_frame_field_flag && ! field_pic_flag). mb_adaptive_frame_field_flag is intax element in the current active SPS and field_pic_flag is a syntax element in th rrent Slice Header. They both are present only if frame_mbs_only_flag is 0. Althor baff_frame_flag is a Slice Header parameter, its value is expected to be the same the slices of a picture. It must be consistent with the mb_adaptive_frame_field_fl e field_pic_flag and the frame_mbs_only_flag settings. This bit is valid only when ig_structure[1:0] indicates the current picture is a frame. Value Name Description FALSE not in MBAFF mode TRUE in MBAFF mode						
	0	FieldPic Field pic to the sa the img a Slice F picture.	: Flag :ture flag ame valu _structur leader p	g, field ue as tl re[1:0] arame	_pic_flag, sp he syntax ele and the frar ter, its value	ecifies the cur ement in the S ne_mbs_only_ e is expected t	rrent slice is a coded field or not.It is set Slice Header. It must be consistent with flag settings. Although field_pic_flag is to be the same for all the slices of a		
		0h	lue	FRAM	E	a slice of a co	oded frame		
		1h		FIELD		a slice of a co	oded field		
5	31	Trellis (Quantiza	ation E	nabled (TQ	Enb)			
		Format				-	Enable		
[ExistsIf]Encode Only		The TQ values for only val	The TQ improves output video quality of AVC CABAC encoder by selecting quantized values for each non-zero coefficient so as to minimize the total R-D cost. This flag is only valid AVC CABAC mode. Otherwise, this flag should be disabled.				ABAC encoder by selecting quantized nimize the total R-D cost.This flag is g should be disabled.		
			Value		Na	me	Description		
		0h			Disable		Use Normal		
	30:28	Trellis (This rou when T(quantize	Juantiza nding sc QEnb is s ed coeffi	ation F cheme set to 1 cients	Rounding (T is only appl L in AVC CA before trun	F QR) ied to the qua BAC mode. Of cating fractior	antized coefficients ranging from 0 to 1 ne of the following values is added to nal part.		



		MF	X_AVC_	IMG_STATE			
27	Trellis This sig TQEnb= TQChro	Trellis Quantization Chroma Disable (TQChromaDisable) This signal is used to disable chroma TQ. To enable TQ for both luma and chroma, TQEnb=1, TQChromaDisable=0. To enable TQ only for luma, TQEnb=1, TOChromaDisable=1.					
26:21	Reserve	ed					
	Format	:			MBZ		
20:17	Reserve	ed					
	Format	:			MBZ		
16	NonFir This sig e.g	stPassFla nals the	ag current pass is	not the first pass. It wi	ll imply designate HW behavior:		
	Value	Name		Descri	ption		
	0h	Disable	Always use th PAK	e MbQpY from initial P	PAK inline object for all passes of		
	1h	Enable	Use MbQpY fi	rom stream-out buffer	if MbRateCtrlFlag is set to 1		
15:13	Reserve	ed					
	Format	:			MBZ		
12	InterM Inter M	bZeroCb B Force C	pFlag - InterN CBP ZERO masl	AB Force CBP to Zero	Control Flag		
	Value	Name		Descri	ption		
	0h	Disable	No Effect				
	1h	Enable	Zero out all A, Confirmance	/C coefficients for the i	inter MB violating Inter		
11:10	MinFra	meWSiz	eUnits				
	This fiel	d is the l	Minimum Fram	e Size Units			
	Value		Name		Description		
	00b	compat	ibility mode	Minimum Frame Size	is in old mode (words, 2bytes)		
	01b	16 byte		Minimum Frame Size	is in 16bytes		
	10b	4Kb		Minimum Frame Size	is in 4Kbytes		
	11b	16Kb		Minimum Frame Size	is in 16Kbytes		
9	MbRate MB Rate	e CtrlFlag e Contro	- MB level R a conformance	ate Control Enabling mask	Flag		
	Value Name Description		ption				
	0h	Disable Apply accumulative delta QP for consecutive passes on top of the macroblock QP values in inline data			secutive passes on top of the		
	1h	Enable	Apply RC QP Buffer except	delta to suggested QP the first pass.	values in Macroblock Status		
	[Programming Net	95		
	This fie	ld is ign	ared when May	croblockStatEnable is o	lisabled or MB level Rate control		
		iu is iyil					



		MF	X_AVC_IMG_STATE				
	flag for the current MB is disable in Macroblock Status Buffer.						
8	Reserve	Reserved					
	Format:		MBZ				
7	Intra/InterMbIpcmFlag - ForceIPCMControlMask This field is to Force IPCM for Intra or Inter Macroblock size conformance mask						
	Value	Nam	e Description				
	0h	Disabl	e Do not change intra macroblocks even.				
	1h	Enable	Change intra macroblocks MB_type to IPCM.				
			Programming Notes				
	This fiel conforn	d is igno nance fla	ored when MacroblockStatEnable is disabled or MB level Intra MB ag for the current MB is disable in Macroblock Status Buffer.				
6:4	Reserve	d					
	Format:		MBZ				
3	FrameSa This is a FrameBi	zUnderf mask bi tRateMi	Flag - FrameBitRateMinReportMask t controlling if the condition of frame level bit count is less than				
	Value	Name	Description				
	0h	Disable	Do not update bit0 of MFC_IMAGE_STATUS control register.				
	1h	Enable	set bit0 and bit 1of MFC_IMAGE_STATUS control register if the total frame level bit counter is less than or equal to Frame Bit rate Minimum limit.				
2	FrameSa	zOverFla	ag - FrameBitRateMaxReportMask				
	This is a FrameBi ⁺	mask bi tRateMa	t controlling if the condition of frame level bit count exceeds x.				
	Value	Name	Description				
	0	Disable	Do not update bit0 of MFC_IMAGE_STATUS control register.				
	1	Enable	Set bit0 and bit 1 of MFC_IMAGE_STATUS control register if the total frame level bit counter is greater than or equal to Frame Bit rate Maximum limit.				
1	InterMb	MaxBit	Flag - InterMBMaxSizeReportMask				
	This is a	mask bi	t controlling if the condition of any inter MB in the frame exceeds				
		MaxSize	Description				
		Value Name Description					
	1	Enable	Set bit0 of MFC IMAGE STATUS control register if the total bit				
			counter for the current MB is greater than the Inter MB Conformance Max size limit.				
0	IntraMb	MaxBit	Flag - IntraMBMaxSizeReportMask				
	This is a IntraMBl	mask bi MaxSize	t controlling if the condition of any intra MB in the frame exceeds				



					G_SIAIE				
		Value	Name		Descrip	otion			
		0h	Disable	Do not update bit0	of MFC_IMAGE_S	TATUS o	control register.		
		1	Enable	set bit0 of MFC_IMA for the current MB i limit.	AGE_STATUS cont s greater than the	rol regis e Intra N	ster if the total bit counter 1B Conformance Max size		
6	31:28	Reserv	ed						
	27:16	InterM	bMaxSz						
[ExistsIf]Encode Only		Forma	t:			U12			
		This fiel for Inte	ld, Inter N r MB	IB Conformance Ma	x size limit,indica	tes the a	allowed max bit count size		
	15:12	Reserv	ed						
		Forma	t:			MBZ			
	11:0	IntraM	bMaxSz						
		Exists I	lf:		//Intra Only				
		Forma	t:		U12				
		This fie size fo	This field, Intra MB Conformance Max size limit, indicates the allowed max bit count size for Intra MB						
7	21.1	Pacarte							
7	0	Reserve	od						
[ExistsIf]Encode Only	Ū	Forma	t:			MBZ			
8	31:24	SliceDe	eltaQpMa	ax[3]					
		Forma	t:				S7		
[ExistsIt]Encode							1		
Chily		Range	: [0:MAX_	QP_DELTA]					
		This field is the Slice level delta QP for total bit-count above FrameBitRateMax - first 1/8 regionThis field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS control register when total bit count for the entire frame exceeds FrameBitRateMax but is within 1/8 of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of (FrameBitRateMax, (FrameBitRateMax+ FrameBitRateMaxDelta>>3).							
	23:16	SliceDe	eltaQpMa	ax[2]					
		Forma	t:			ι	8L		
		Range	: [0:MAX_	QP_DELTA]					
		This fie and be MFC_II betwee	eld is the elow 1/ 4 MAGE_ST en 1/8 an	Slice level delta QP f This field is used to ATUS control registe d ¼ of FrameBitRate	or bit-count above calculate the sugger when total bit of MaxDelta above	ve Frame gested s count fo FrameB	eBitRateMax - above 1/8 lice QP into the r the entire frame is itRateMax, i.e., in the		



		MFX_AVC_IMG_STATE					
		range of ((FrameBitRateMax+ FrameBitRateMaxDelta>>3), (F FrameBitRateMaxDelta>>2).	rameBitRateMax+				
	15:8	SliceDeltaOpMax[1]					
		Format:	S7				
		Range: [0:MAX_QP_DELTA]					
		This field is the Slice level delta QP for bit-count above Frame and below 1/2 This field is used to calculate the suggested sl MFC_IMAGE_STATUS control register when total bit count fo between ¼ and ½ of FrameBitRateMaxDelta above FrameBit range of ((FrameBitRateMax+ FrameBitRateMaxDelta>>2), (F FrameBitRateMaxDelta>>1).	eBitRateMax - above1/ 4 ice QP into the r the entire frame is RateMax, i.e., in the FrameBitRateMax+				
	7:0	SliceDeltaQpPMax[0]					
		Format:	S7				
		Range: [0:MAX_QP_DELTA]					
		This field is the Slice level delta QP for bit-count above Frame 2This field is used to calculate the suggested slice QP into th control register when total bit count for the entire frame is al more than half the distance of FrameBitRateMaxDelta , i.e., ir ((FrameBitRateMax+ FrameBitRateMaxDelta > 1), infinite).	eBitRateMax - above 1/ e MFC_IMAGE_STATUS bove FrameBitRateMax by n the range of				
9	31:24	SliceDeltaQpMin[3]					
		Format:	S7				
[ExistsIf]Encode Only							
Citiy		Range: [0:MAX_QP_DELTA]					
		This field is the Slice level delta QP for total bit-count below FrameBitRateMin - fin 1/8 regionThis field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS control register when total bit count for the entire frame is I than FrameBitRateMin and greater than or equal to 1/8 the distance of FrameBitRateMinDelta from FrameBitRateMin, i.e., in the range of [(FrameBitRateMi FrameBitRateMinDelta >>3), FrameBitRateMin).					
	23:16	SliceDeltaQpMin[2]					
		Format:	S7				
		Range: [0:MAX_OP_DELTA]					
		This field is the Slice level delta OP for hit-count below Frame	eBitRateMin - below 1/8				
		and above 1/ 4This field is used to calculate the suggested sl MFC_IMAGE_STATUS control register when total bit count fo between one-eighth and quarter the distance of FrameBitRate FrameBitRateMin, i.e., in the range of [(FrameBitRateMin- Fra (FrameBitRateMin- FrameBitRateMinDelta>>3)).	lice QP into the r the entire frame is teMinDelta from meBitRateMinDelta>>2),				
	15:8	SliceDeltaQpMin[1]					
		Format:	S7				



MFX	AVC	IMG	STATE

		Range:	[0:MAX_	QP_DELTA]						
		This fiel and abo MFC_IN betwee FrameB (FrameB	This field is the Slice level delta QP for bit-count below FrameBitRateMin- below 1/4 and above 1/ 2This field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS control register when total bit count for the entire frame is between quarter and half the distance of FrameBitRateMinDelta from FrameBitRateMin, i.e., in the range of [(FrameBitRateMin- FrameBitRateMinDelta >>1), (FrameBitRateMin- FrameBitRateMinDelta >>2)).							
	7:0	SliceDe	ltaQpMi	n[0]						
		Format			S7					
		r								
		Range:	[0:MAX_	QP_DELTA]						
	This field is the Slice Level Delta QP for bit-count below FrameBitRateMin - 2This field is used to calculate the suggested slice QP into the MFC_IMAGE control register when total bit count for the entire frame is below FrameBit more than half the distance of FrameBitRateMinDelta , i.e., in the range of (FrameBitRateMin- FrameBitRateMinDelta > 1).									
10	31	FrameB	itrateMa	axUnit						
[Evicte]f]Encodo		This field	d is the F	rame Bitrate N	Aaximum Limit Units.					
Only		Value	Name	E D'ID ()	Description					
- ,		0	Byte	FrameBitRatel FrameBitrateN FrameBitrateN	Max is in units of 32 Bytes when AaxUnitMode is 1 and in units of 128 Bytes if AaxUnitMode is 0					
		1	Kilo Byte	FrameBitRatel FrameBitrateN FrameBitrateN	Max is in units of 4KBytes Bytes when AaxUnitMode is 1 and in units of 16KBytes if AaxUnitMode is 0					
	30	FrameB	itrateMa	axUnitMode						
		This field	d is the F	rame Bitrate N	Aaximum Limit Units.					
		Value	I	Name	Description					
		0h	compat	ibility mode	FrameBitRateMaxUnit is in old mode (128b/16Kb)					
		1h	New mo	ode	FrameBitRateMaxUnit is in new mode (32byte/4Kb)					
	29:16	FrameBitRateMax This field is the Frame Bitrate Maximum Limit. This field along with FrameBitrateMaxUnit determines maximum allowed bits in a frame before multi- gets triggered (when enabled). In other words, multi-pass is triggered when the frame byte count exceeds this value. When FrameBitrateMaxUnitMode is 0(compatibility mode) bits 16:27 should be used, bits 28 and 29 should be 0								
		Value	e Nam	e	Description					
		0-512K	В	The progra 0.	mmable range is 0-512KB when FrameBitrateMaxUnit is					
		0- 8190KB		The progra is 1.	mmable range is 0-8190KB when FrameBitrateMaxUnit					
	15	FrameB	itrateMi	nUnit						



			MF	X_A	VC_	IMG_STATE	
		This field	d is the l	Frame B	Bitrate N	1inimum Limit Units.	
		Value	Name			Descri	ption
		0	Byte	Framel Framel	BitRate Bitrate	Max is in units of 32 By 1inUnitMode is 1 and i	ytes when in units of 128 Bytes if
				Framel			
		1	Kilo Byte	FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0			
	14	FrameB This field	rameBitrateMinUnitMode				
		Value		Name			Description
		0h	Compa	tibility r	node	FrameBitRateMaxUni	t is in old mode (128b/16Kb)
		1h	New m	ode		FrameBitRateMaxUni	t is in new mode (32byte/4Kb)
	13:0	FrameB RangeTh 0.Progra Frame B minimur other wo value. W used, bit	itRateN ne progr immable itrate M n allowe ords, mu fhen Fra ts 12 and	RateMin e programmable range 0-512KB When FrameBitrateMinUnit is in nmable range is 0-8190 KB when FrameBitrateMinUnit is in 1.This field is the trate Minimum Limit ()This field along with FrameBitrateMinUnit determines allowed bits in a Frame before Multi-Pass gets triggered (when enabled). In rds, multi-pass is triggered when the actual frame byte count is less than this nen FrameBitrateMinUnitMode is 0 (compatibility mode) bits 0:11 should be s 12 and 13 should be 0			
11	31	Reserve	d				
[Eviate][f]Eviate][a	30:16	FrameB	itRateN	laxDelt	a		
[Existsif]Encode Only		Format					U15
Ciny		This field shares the <u>0(compare</u>)	d is usec ne same atibility i	l to sele Framel mode) k	ect the s BitrateM pits 16:2	lice delta QP when Fra 1axUnit. When FrameB 17 should be used, bits	ameBitRateMax Is exceeded. It BitrateMaxUnitMode is 5 28, 29 and 30 should be 0
		Valu	e N	ame		Des	scription
		0-1024	KB		The Pro Frame	ogrammable range 0-: BitRateMaxUnit is 0.	1024KB when
		0- 16380K	В		The Pro Frame	ogrammable range is (BitRateMaxUnit is 1.	0-16380KB when
		0h	[De	efault]			
	15	Reserved					
		Format: MBZ					
	14:0	FrameBitRateMinDelta					
		Range: Prograr	The pro nmable	gramm range i	able ran s 0-1638	ige 0-1024KB When Fi 80KB when FrameBitra	rameBitrateMinUnit is in 32Bytes. ateMinUnit is in 4Kbytes.
		This fiel when F FrameB	d is use rameBitl itrateMi	d to sel RateMir nUnitM	ect the n Is exce lode is (slice delta QP eeded. It shares the sau)(compatibility mode)	me FrameBitrateMinUnit. When bits 0:11 should be used, bits



		MFX_AVC_IMG_S	TATE		
		12, 13 and 14 should be 0.Note: HW requi FrameBitRateMinDelta <= 2*FrameBitRate unpredicted behavior.	res the foll MinMust b	owing condition be true, otherwise it may cause	
12	31:21	Reserved			
		Format:		MBZ	
	20	Reserved			
		Format:		MBZ	
	19	Reserved			
		Format:		MBZ	
	18:16	Reserved			
		Format:		MBZ	
	15:0	Reserved			
		Format:		MBZ	
13	31:30	Reserved			
		Format:		MBZ	
	29	Current Picture Has Performed MMCO5			
		Set to 1 if the current Pic has performed the memory_management_control_operation = = 5.			
	28:24	Number of Reference Frames			
		Format:		U5	
		Range: Range 0 to MaxDpbSize (=16 for Level 4.1)			
		Specifies the maximum number of referen existed in the current DBP for decoding th	ce frames (e current p	frames, field pairs, unpaired field) icture.	
	23:22	Reserved			
		Format:		MBZ	
	21:16	Number of Active Reference Pictures fro	m L1		
		Format:	U	16-1	
	Specifies the initial maximum reference index value minus 1 to access t List. It is extracted from PPS. It corresponds to the number of active ref from L1 to decode the current picture. It can be modified by the slice h			inus 1 to access the L1 Reference mber of active reference pictures fied by the slice header if B picture.	
		Value		Name	
		[0,31]			
	15:14	Reserved			
		Format:		MBZ	
	13:8	Number of Active Reference Pictures fro	m L0		
		Format:	U	16-1	
		Specifies the initial maximum reference index value minus 1 to access the L0 Reference			

Г



		MFX_AV	C_IMG_S	TATE		
		List. It is extracted from PPS. It corresponds to the number of active reference pictures from L0 to decode the current picture. It can be modified by the slice header if num_ref_idx_active_override_flag is set. Valid for both P and B pictures.				
		Value Name			Name	
		[0,31]	[0,31]			
	7:0	Initial QP Value				
		Format:			S7	
				·		
			Descripti	on		
		Range: [-26,25]				
		Short Format Only				
		Initial QP value for a Slice, slice_qp_delta in slice hea	, extracted from I der and mb_qp_c	PPS. It may further g delta in MB header.	get modified by	
14	31:24	24 Log2_max_pic_order_cnt_lsb_minus4				
		Exists If:	//Short Format (Only		
[ExistsIf] Short		It is a SPS syntax element, used to determine how many bits in the b				
i offiat offiy		to represent pic_order_cnt_lsb syntax element in the slice header. Unsigned			er. Unsigned	
	23:16	Log2_max_frame_num_m	ninus4			
		Exists If:	//Short Format	Only		
		It is a SPS syntax element, used to determine how many bits in the bitstream are used				
		to represent frame_num sy	ntax element in	the slice header. Un	signed.	
	15	deblocking_filter_control	_present_flag			
		Exists If: //Short Format Only				
		It is a PPS syntax element, indicates if more deblocking filter control syntax elements are present in the slice header.				
	14:12	num_slice_groups_minus	1			
		Exists If:	//Short Format	Only		
		BitField It is a PPS syntax element.Use for Slice Header parsing only, to read in				
		slice_group_change_cycle,	if any, but is not	used by H/W, i.e. no	o slice group	
		support.Desc				
	11	redundant_pic_cnt_prese	nt_flag			
		Exists If:	//Short Format (Only		
		It is a PPS syntax element.	Jse for Slice Head	der parsing only, to	read-in	
		redundant_pic_cnt, if any, l processing.	but is not used by	y H/W, i.e. no suppo	ort for redundant slice	
	10:8	slice_group_map_type				
		Exists If:	//Short Format	Only		



		MFX_AV	/C_IMG_STATE	
		It is a PPS syntax element.Use for Slice Header parsing only, to read in slice_group_change_cycle, if any, but is not used by H/W, i.e. no slice group support. 4 Reserved		
	7:4			
		Format:	MBZ	
		IDR flag is decoded from NAL Header Byte		
	3:2	Pic_order_cnt_type		
		Exists If: //Short Format Only		
		It is a SPS syntax element	Use for Slice Header parsing only.	
	1	Delta_pic_order_always_	zero_flag	
		Exists If:	//Short Format Only	
		It is a SPS syntax element	Use for Slice Header parsing only.	
	0	Pic_order_present_flag		
		Exists If:	//Short Format Only	
		It is a PPS syntax element	Use for Slice Header parsing only.	
15	31:16	Curr Pic Frame Num		
		Exists If:	//Short Format Only	
[ExistsIf] Short		Format:	U16	
r onnat only		Derived from Slice Header syntax element		
	15:0	Slice Group Change Rate	e	
		Exists If:	//Short Format Only	
		Format:	U16-1	
		It is a PPS syntax element		
		Use for Slice Header parsing only, to read in slice_group_change_cycle, if any,		
		used by H/W, i.e. no slice	group support.	
16	31:0	Reserved		
[ExistsIf]: Short				
Format only				



MFX_AVC_REF_IDX_STATE

Source: VideoCS

Length Bias:

This is a slice level command and can be issued multiple times within a picture that is comprised of multiple slices. The same command is used for AVC encoder (PAK mode) and decoder (VLD mode); it is not need in decoder IT mode.

The inline data of this command is interpreted differently for encoder as for decoder. For decoder, it is interpreted as RefIdx List L0/L1 as in AVC spec., and it matches with the DXVA2 AVC API data structure for decoder in VLD mode: RefPicList[2][32] (L0:L1, 0:31 RefPic). But for encoder, it is interpreted as a Reference Index Mapping Table for L0 and L1 reference pictures. For packing the bits at the output of PAK, the syntax elements must follow the definition of RefIdxL0/L1 list according to the AVC spec. However, the decoder pipeline was designed to use a variation of that standard definition, as such a conversion (mapping) is needed to support the hardware design.

The Reference lists are needed in processing both P and B slice in AVC codec. For P-MB, only L0 list is used; for B-MB both L0 and L1 lists are needed. For a B-MB that is coded in L1-only Prediction, only L1 list is used.

Programming Notes

DXVA2 specifies that an application will create the RefPicList L0 and L1 and pass onto the driver. The content of each entry of RefPicList L0/L1[] is a 7-bit picture index. This picture index is the same as that of RefFrameList[] content. This picture index, however, is not defined the same as the frame store ID (0 to 16, 5-bits) we have implemented in H/W. Hence, driver is required to manage a table to convert between DXVA2 picture index and the frame store ID. As such, the final RefPicList L0/L1[] that the driver passes onto the H/W is not the same as that defined in the DXVA2.

DWord	Bit	Description			
0	31:29	Command Type			
		Default Value:	3h PARALLEL_VIDEO_P	IPE	
		Format:	OpCode		
	28:27	Pipeline			
		Default Value:	2h MFX_AVC_REF_IDX_ST	ATE	
		Format:	OpCode		
	26:24	Command Opcode			
		Default Value:		1h AVC	
		Format:		OpCode	
	23:21	SubOpcodeA			
		Default Value:	0h MFX_AVC_REF_IDX_ST	ATE	
		Format:	OpCode		
	20:16	SubOpcodeB			
		Default Value:	4h MFX_AVC_REF_IDX_ST	ATE	
		Format:	OpCode		
	15:12	Reserved			
		Format:		MBZ	



			M	FX_AVC_REF_IDX_ST	ATE		
	11:0	DWord	l Length				
		Defaul	t Value:		0008h		
		Forma	t:		=n		
		Excludes DWords 0,1					
1	31:1	Reserv	ed				
		Forma	t:		MBZ		
	0	RefPicList Select Num_ref_idx_l1_active is resulted from the specifications in both PPS and Slice Header for the current slice. However, since the full reference list L0 and/or L1 are always sent, only present flags are specified instead. This parameter is not present in the DXVA					
		Value	Name	Desci	iption		
		0	RefPicList 0	The list that followed represents RefL Mapping Table L0 (Encoder PAK mod	ist L0 (Decoder VLD mode) or Ref Idx e)		
		1	RefPicList1	The list that followed represents RefL Mapping Table L1 (Encoder PAK mod	ist L1 (Decoder VLD mode) or Ref Idx e)		
29	31:0	Refere	nce List Entr	у			
		It always specifies the full 32 reference pictures in the selected list, regardless they are "existing picture" or not. If a picture is non-existing, the corresponding entry should be set to all ones. Each list entry is 1 byte. A 32-bit DW can hold 4 list entries in the following format 31:24 entry X+3 (e.g. listY_3) 23:16 entry X+2 (e.g. listY_2) 15:8 entry X+1 (e.g. listY_1) 7:0 entry X (e.g. listY_0) 					
		 X is replaced by the paddr[2:0] * 4 ; paddr[5:0] with 0x20 and 0x27, and Y is replaced by 0 or 1. The byte definition for a reference picture : Bit 7: Non-Existing - indicates that frame store index that should have been at this entry did not exist and was replaced by an index 0 (a valid entry) for error concealment Bit 6: Long term bit - set this reference picture to be used as long term reference Bit 5: Field picture flag - indicates frame/field Bit 4:0: Frame store index or Frame Store ID (Bit 4:1 is used to form the binding table index in this implementation) 					



MFX_AVC_REF_IDX_STATE					
	This is the final Reference List L0 or L1 after any reordering specified in the Slice Header as well as modified by the driver, and its indices values are all translated to this specification. If the reference picture is a frame (Bit5 = 1), frame store ID is always an even number. This list is used in outputting MV information by the BSD unit in VLD mode. DMV access also reads and writes Mvlist0 using this frame store ID. If this set of fields is interpreted as Reference Index Mapping Table L0/L1, the same field alignment is followed, i.e. 4 mapping entries per DW. Each mapping entry is one byte in size, but only the least significant 5 bits [4:0] is relevant. Driver should zero all the upper bits [7:5] for each entry.				



MFX_AVC_SLICE_STATE

Source: VideoCS

2

Length Bias:

This is a slice level command and can be issued multiple times within a picture that is comprised of multiple slices. The same command is used for AVC encoder (PAK mode) and decoder (VLD and IT modes).

Programming Notes

MFX_AVC_SLICE_STATE command is not issued for AVC DXVA2 Short Format Bitstream decode, instead MFD_AVC_SLICEADDR command is executed to retrieve the next slice MB Start Address X and Y by H/W itself.

Dword	ΒΙζ	Description					
0	31:29	Command Type					
		Default Value:	3h PARALLEL_VIDEO_	PIPE			
		Format:	OpCode				
	28:27	Pipeline					
		Default Value:	2h MFX_AVC_SLICE_S	ΤΑΤΕ			
		Format:	OpCode				
	26:24	Command Opcode					
		Default Value:		1h AVC			
		Format:		OpCode			
	23:21	SubOpcodeA					
		Default Value:	0h MFX_AVC_SLICE_S	ТАТЕ			
		Format:					
	20:16	Command SubOpcodeB					
		Default Value:	ΤΑΤΕ				
		Format:	OpCode				
	15:12	Reserved					
		Format:		MBZ			
	11:0	DWord Length					
		Default Value:	8h DWORD_COUN	NT_n			
		Format:	=n				
		Excludes DWords 0,1					
1	31:4	Reserved					
		Format:		MBZ			
3:0 Slice Type It is set to the value of the syntax element read f			lement read from the	Slice Header.			
		Value		Name			
		0000b		P Slice			
		0001b		B Slice			
		0010b		I Slice			



		MFX_AVC_SLICE	E_STATE					
		0011b-1111b	Reserved					
		Programming Notes						
		Bits[3:2] must be 0						
2	31:30	Reserved]				
		Format:	MBZ					
	29:24	Number of Reference Pictures in Inter-predi	iction List 1					
		Format:	U6					
		in the reference list L1; otherwise (if Slice Type This field can be derived for a B Slice from the NumRefIdxActiveMinus1 as, Num_Ref_Idx_L1 =	is not a B Slice), this field must be set to 0. Slice Header syntax element NumRefIdxActiveMinus1[1] + 1.	.ry				
		Value	Name					
		0-32						
	23:22	Reserved	·					
		Format:	MBZ					
	21:16	Number of Reference Pictures in Inter-prediction List 0						
		Format: U6						
		This field is valid for encoding a P or B Slice, for which it is expected to have at least one entry in the reference list L0; otherwise (if Slice Type is not a P or B Slice), this field must be set to 0. This field can be derived for a P or B Slice from the Slice Header syntax element NumRefIdxActiveMinus1 as, Num_Ref_Idx_L0 = NumRefIdxActiveMinus1[0] + 1.						
		Value	Name					
		0-32						
	15:11	Reserved						
		Format:	MBZ					
	10:8	Log 2 Weight Denom Chroma						
		Format:	U3					
		Value	Name					
		0-7						
	7:3	Reserved						
		Format:	MBZ					
	2:0	Log 2 Weight Denom Luma						
		Format:	U3					
		It is the base 2 logarithm of the denominator for	or all Luma weighting factors.					
		It is set to the value of the syntax element read	trom the Slice Header Pred_Weight_Table().					
			Name					
2	21.20							
3	31:30	weighted Prediction Indicator						



		I	MFX_AVC_S	SLICE_STATE	
	This field indicates the Weighted Prediction mode for a P or B Slice. It is a combined field corresponding to the syntax element WeightedBiPredIdc or WeightedPredFlag read from the current active PPS.				
	• 1	• If it is a B-Slice, these bits are interpreted as:			
	00b - 5 01b - 5 10b - 5 11b - F	00b - Specifies the default weighted inter-prediction to be applied 01b - Specifies the explicit weighted inter-prediction to be applied 10b - Specifies the implicit weighted inter-prediction to be applied 11b - Reserved (not allowed)			
	• 1	if it is a P S	lice, these bits are int	erpreted as:	
	00b - E 01b - E 10b - 1	Disables we Enables we L1b - Reser	eighted inter-predictio ighted inter-predictio ved	on (Default weighted) on (Explicit weighted)	
			P	rogramming Notes	
	Only w L1 and comma Only w	/hen in B S //or a L0 we and. when in P S	lice with Weighted_Pr eight+offset tables be Slice with Weighted_P	red_Idc = 1 (explicit weighted prediction), will there be a eing sent to the BSD unit through the Slice_State red_Idc = 1, will there be a L0 weight+offset table being	
	If Weig comma ignore	o the BSD. ghted_Pred and should d.	_Idc != 1 for B Slice o I be issued to send th	r Weighted_Pred_Idc =0 for P Slice, no Slice_State ese tables. If still being issued, the data is read but	
	DXVA flags a	specifies W re combine	/eighted_Bipred and ` ed and specified in sli	Weighted_Pred in frame-level state. However, these two ce level for both P and B slice type.	
29	Direct Type of otherwi	Prediction direct pre ise, it must	Type diction used for B Slic be set to 0.	ces. This field is valid only for Slice_Type = B Slice;	
		V	alue	Name	
	0			Temporal	
	1			Spatial	
28:27	7 Disable	e Deblocki	ng Filter Indicator		
	Value	Name		Description	
	00b		FilterInternalEdgesFl	ag is set equal to 1	
	01b		Disable all deblockir read; filterInternalEd	ng operation, no deblocking parameter syntax element is gesFlag is set equal to 0	
	10b		Macroblocks in diffe filterInternalEdgesFla	rent slices are considered not available; ag is set equal to 1	
	11b	Reserved	Not defined in AVC		
26	Reserv	ed			



		MFX_4	AVC_SLICE	STA	re i i i i i i i i i i i i i i i i i i i			
		Format: MBZ						
	25:24	Cabac Init Idc[1:0] Specifies the index for determining the initialization table used in the context variable initialization process.						
		Value			Name			
		0-2						
			Program	ming Notes	5			
		Cabac initialization is also d current SliceQP value.	ependent on the fi	eld/frame p	victure type, Slice type, and the			
	23:22	Reserved		l.				
		Format:			MBZ			
	21:16	Slice Quantization Parameter Quantization Parameter for current slice. Derived from PPS and slice_delta_qp syntax element in Slice Header. It is needed for CABAC context initialization and deblocking filter control. And it is also used as the starting QP value in the very first MB of a slice. It is in the range of unsigned integer 0 to 51. for 8-bit pixel bit-depth.						
	15:12	Reserved						
		Format:			MBZ			
	11:8	Slice Beta Offset Div2						
		Format:	S3 2's Compleme	nt				
		Range: [-6, 6] Inclusive						
		Specifies the offset used in accessing the deblocking filter strength tables.						
	7:4	Reserved						
		Format:			MBZ			
	3:0	Slice Alpha C0 Offset Div2						
		Format:	S3 2's Compleme	nt				
		Range: [-6, 6] Inclusive						
		Specifies the offset used in accessing the deblocking filter strength tables.						
4	31:24	Slice Vertical Position This field specifies the position macroblocks. The fields (Slice_MB_Start_H only. They are ignored by ha position is provided by the p Derived	on in y-direction of or_Pos, Slice_MB_S rdware in decoding per-macroblock obj	f the first m Start_Vert_P g IT mode a ect comma	acroblock in the Slice in unit of os) are valid in VLD (decoding) mode and encoding mode (whereas the nd).			
			Program	ming Notes	5			
		Error Handling: Driver need	s to check if FirstM	bY starts at	0 on the first slice of frame. If not,			



		MFX_AVC_SLIC	CE_STATE			
		driver needs to add a phantom slice with FirstMbX and FirstMbY set to 0.				
	23:16	Slice Horizontal Position This field specifies the position in x-direction of the first macroblock in the Slice in unit of macroblocks. Derived				
		Progra	mming Notes			
		Error Handling: Driver needs to check if Firs driver needs to add a phantom slice with Fi	tMbY starts at 0 on the first slice of frame. If not, rstMbX and FirstMbY set to 0.			
	15	Reserved				
		Format:	MBZ			
	14:0	Slice Start Mb Num				
		Exists If: //Decode	r Only			
		The MB number (linear MB address in a pictor Slice Horizontal Position (Slice_MB_Start_Hor (Slice_MB_Start_Vert_Pos) in the picture.	ure) at the start of a Slice, it must match with the r_Pos) and Vertical Position			
		Programming Notes				
		In creating the Phantom Slice for error conc of MB in the current picture + 1.	ealment, this field should set to the total number			
5	31:24	Reserved				
		Format:	MBZ			
	23:16	6 Next Slice Vertical Position This field specifies the position in y-direction of the first macroblock in the next Slice in unit macroblocks. This field is primarily used for error concealment. In the case that current slice is the last slice this field is primarily used for error concealment. In the case that current slice is the last slice				
	15:8	Reserved				
		Format: MBZ				
	7:0	Next Slice Horizontal Position This field specifies the position in x-direction of the first macroblock in the next Slice in unit of macroblocks. This field is primarily used for error concealment. In the case that current slice is the last slice, this field should set to 0.				
6 Encoder Only	31	Rate Control Counter Enable To enable the accumulation of bit allocation for rate control This field enables hardware Rate Control logic. The rest of the RC control fields are only value when this field is set to 1. Otherwise, hardware ignores these fields				
		Value	Name			
		0	Disable			
		1	Enable			
	30	ResetRateControlCounter To reset the bit allocation accumulation court	nter to 0 to restart the rate control.			



	Value					N	ame			
	0				Not Res	et				
	1				Reset					
29:28	RC Triggle Mode									
	Value	Nan	ne			Descriptio	on			
	00b	Always Ra Control	te	Whereas RC becomes active if sum_act > sum_target or sum_ac sum_target						
	01b Gentle Rate Control			whereas RC becomes active if sum_act > upper_midpt or sum_a lower_midpt						
	10b	Loose Rat Control	e	whereas RC becomes active if sum_act > sum_max or sum_act sum min						
	11b Reserved									
27:24	RC Sta	ble Tolera	nce							
	Forma	ət:					U4			
	This fie	eld specifies	the tole	rance requi	red to de	activate RC once it	has been triggered.			
			Value				Name			
	0-15									
	contro	is what type	Value	c behavior i	Name					
	0				Disable					
	Enable									
22	RC Par	nic Type	aturaan t		ic motho	de				
					ic metho	us N	ame			
					OP Panic					
	1				CBP Panic					
	Programming Notes									
	If it is set to 0, in panic mode, the macroblock QP is maxed out, setting to requested QP + QP_max_pos_mod. If it is set to 1, for an intra macroblock, AC CBPs are set to zero (note that DC CBPs are not modified). For inter macroblocks, AC and DC CBPs are forced to zero.									
21	MB Type Direct Conversion Disable									
	Exists If: //B-Slice									
	For all	For all Macroblock type conversions in different slices, refer to Section "Macroblock Type Conversion Rules" in the same volume.								
	Conve	rsion Rules"	in the sa	ame volume	e.	t silces, relef to set				



			MF	X_AVC_SLICE_STATE					
	0		Enab	Enable direct mode conversion					
	1		Disab	Disable direct mode conversion					
	Programming Notes								
	This field is zero for all other slices other than B-Slice.								
20	MB Type Skip Conversion Disable								
	Exists If: //P-Slice or B-Slice								
	For all Macroblock type conversions in different slices, refer to Section "Macroblock Type Conversion Rules" in the same volume.								
	V	/alue		Name					
	0		Ena	Enable skip type conversion					
	1		Disa	Disable skip type conversion					
	Programming Notes								
	I his field is zero for all other slices other than P_Slice or B-Slice. \								
19	Is Last Slice								
	Valu	e I	Name	ame Description					
	1			Current slice is the last slice of a picture					
	0			Current slice is NOT the last slice of a picture					
18	Reserved								
17	Header Insertion Present in Bitstream								
	Value	Name		Description					
	0		No header insertion into the output bitstream buffer, in front of the current						
	1	insertion into the output bitstream buffer is present, and is in front of							
	the current slice encoded bits.								
16	SliceData Insertion Present in Bitstream								
	Value	Nam	le	Description					
	0		No S	No Slice Data insertion into the output bitstream buffer					
	1			Slice Data insertion into the output bitstream buffer is present.					
15	Tail Insertion Present in bitstream								
	Value	Name	Description						
	0		No tail insertion into the output bitstream buffer, after the current slice encoded bits						
	1		Tail insertion into the output bitstream buffer is present, and is after the current slice encoded bits.						
14	Reserved								
	Format: MBZ								



				M	FX_AVC_SLI		TE				
	13	EmulationByteSliceInsertEnable To have PAK outputting SODB or EBSP to the output bitstream buffer									
			Value		Name		Description				
		0				outputting R	BSP				
		1				outputting El	BSP				
	12	CabacZeroWordInsertionEnable									
		To pad the end of a SliceLayer RBSP to meet the encoded size requirement.									
		Value	Name		Description						
		0		No Ca							
		1		Allow internal Cabac_Zero_Word generation and append to the end of RBSP (effectively can be used as an indicator for last slice of a picture, if the assumption is only the last slice of a picture needs to insert CABAC_ZERO_WORDs.							
	11:8	Reserve	ed								
		Format	t:				MBZ				
		For SliceID extension.									
	7:4	Slice ID [3:0] To identify the output data (coding information record) returned for rate control from PAK to ENC and VPP.									
	3:2	Reserved									
		Format	t:				MBZ				
		For StreamID extension.									
	1:0	Stream ID [1:0] To identify the output data (coding information record) returned for rate control from PAK to ENC and VPP.									
7	31:29	Reserve	ed								
		Format	t:				MBZ				
Encoder	28:0	Indirec	Indirect PAK-BSE Data Start Address (Write)								
Only		Exists I	f:		//AVC Encode	e Mode					
		This field specifies the memory starting address (offset) to write out the compressed bitstream data from the BSE processing. This pointer is relative to the MFC Indirect PAK-BSE Object Base Address. It is a byte-aligned address for the AVC bitstream data in both CABAC/CAVLC Modes. For Write, there is no need to have a data length field. It is assumed the global memory bound									
		check specified in the IND_OBJ_BASE_ADDRESS command (Indirect PAK-BSE Object Access									
					Value		Name				
		0 - 512									
		0 512									
8	31:24	Magnit	ude of	QP Ma	x Negative Modifie	r					


		MFX_	AVC_SLICE_	STATE							
Encoder		This field specifies the lower	limit of the QP modi	fier.							
Only		Value			Name						
		0-51									
	23:16	Magnitude of QP Max Positive Modifier									
		Format:			U8						
		This field specifies the upper	This field specifies the upper limit of the QP modifier.								
		Value		Name							
		0 - 15									
	15:12	Shrink Param - Shrink Resi	stance								
		Format:			U4						
		This field specifies the additi	onal points added ea	ach time decreas	ed correction is invoked.						
		Value			Name						
		0 - 15									
	11:8	Shrink Param - Shrink Init									
		Format: U4									
		This field specifies the initial points required to trip decreased control.									
		Value			Name						
		0 - 15									
	7:4	Grow Param - Grow Resistance									
		Format:			U4						
		This field specifies the additi	onal points added ea	ach time increase	ed correction is invoked.						
		Value			Name						
		0 - 15									
	3:0	Grow Param - Grow Init									
		Format:			U4						
		This field specifies the initial	points required to tr	ip increased con	trol.						
		Value			Name						
		0 - 15									
9	31	RoundInterEnable									
		Format:		Enable							
Encoder Only		When this bit is not set, Rou	ndInter defaults to 2.								
	30:28	RoundInter									
		Format:			U3						
		Rounding precision for Inter	quantized coefficien	ts							
		Value		Name							
		000b	+1/16 [Default]								
		0011	12/16								
		0100	+2/10								



	011b	+4/16						
	100b	+5/16						
	101b	+6/16						
	110b	+7/16						
	111b	+8/16						
27	RoundIntraEnable							
	Format:		Enable					
	When this bit is not	set, RoundIntra defaults t	o 4.					
26:24	RoundIntra							
	Format:		U3					
	Rounding precision	for Intra quantized coeffic	cients					
	Value		Name					
	000b	+1/16 [Default]						
	001b	+2/16						
	010b	+3/16	+3/16					
	011b	+4/16						
	100b	+5/16	+5/16					
	101b	+6/16	+6/16					
	110b	+7/16	+7/16					
	111b	+8/16						
23:20	Correct 6							
	Format:		U4					
	This field specifies th	e points used in the lowe	ermost RC region when sum_act <= sum_m					
		Value	Name					
	0 - 15							
19:16	Correct 5							
	Format:		U4					
	This field specifies th lower_midpt.	e points used in the fifth	RC region when sum_act > sum_min but <					
		Value	Name					
	0 - 15							
15:12	Correct 4							
	Format:		U4					
	This field specifies th sum_target.	e points used in the four	th RC region when sum_act > lower_midpt					
		Value	Name					
	0 - 15							
11.0	Correct 2							



		MFX_AVC_SLICE	_STATE							
		Format:	U4							
		This field specifies the points used in the third RC region when sum_act > sum_target but								
		upper_midpt.								
		Value	Name							
		0 - 15								
	7:4	Correct 2								
		Format:	U4							
		This field specifies the points used in the second RC region when sum_act > upper_mid <= sum_max.								
		Value	Name							
		0 - 15								
	3:0	Correct 1								
		Format:	U4							
		This field specifies the points used in the topmo	st RC region when sum_act > sum_max.							
		Value	Name							
		0 - 15								
10	31:28	ClampValues - CV7								
Encodor	27:24	CV6								
Only	23:20	CV5								
- ,	19:16	CV4								
	15:12	CV3								
	11:8	CV2								
	7:4	CV1								
	3:0	CV0 - Clamp Value 0								
		Format:	U4							
		If the magnitude of coefficients at locations	assigned with CV0 (mapping shown below)							
		exceeds 2^{cvo} -1, they are replaced with 2^{cvo} -1	For coefficients at locations marked as							
		and chroma blocks subblocks containing A	C coefficiencts (blocks) sublocks with only							
		DC coeffs will not be clamped).	e coefficiencis (blocks (sublocks with only							
		For 4x4 frame block, each coefficient is r	napped to one of the eight CV values as							
		following:								
		none CV7 CV5 CV4								
		CV7 CV6 CV4 CV3								
		CV5 CV4 CV2 CV1								
		CV4 CV3 CV1 CV0								
		For 8x8 frame block, each coefficient is r	napped to one of the eight CV values as							
		following:								
		none none CV7 CV6 CV5 CV4 CV3 CV3								



				MF	X_A		:_S	LIC	E_STATE
r	none	CV7	CV6	CV5	CV4	CV3	CV3	CV2	
(CV7	CV6	CV5	CV4	CV3	CV3	CV2	CV2	
	CV6	CV5	CV4	CV3	CV3	CV2	CV2	CV1	
(CV5	CV4	CV3	CV3	CV2	CV2	CV1	CV1	
0	CV4	CV3	CV3	CV2	CV2	CV1	CV1	CV0	
0	CV3	CV3	CV2	CV2	CV1	CV1	CV0	CV0	
(CV3	CV2	CV2	CV1	CV1	CV0	CV0	CV0	
F	or 4x ollow	(4 fie /ing:	ld blo	ock, e	each	coef	ficier	nt is m	napped to one of the eight CV values as
r	none	CV6	CV3	CV1					
C	CV7	CV6	CV3	CV1					
C	CV5	CV4	CV2	CV0					
(CV5	CV4	CV2	CV0					
F fe	or 8x	(8 fie /ing:	ld blo	ock, e	each	coef	ficier	nt is m	napped to one of the eight CV values as
r	none	none	CV6	CV5	CV4	CV3	CV2	CV1	
r	none	CV7	CV6	CV5	CV4	CV3	CV2	CV1	
C	CV7	CV6	CV5	CV4	CV3	CV3	CV2	CV1	
C	CV7	CV6	CV5	CV4	CV3	CV2	CV2	CV1	
(CV6	CV5	CV4	CV4	CV3	CV2	CV1	CV0	
(CV6	CV5	CV4	CV3	CV2	CV2	CV1	CV0	
C	CV5	CV5	CV4	CV3	CV2	CV1	CV1	CV0	
0	CV5	CV5	CV4	CV3	CV2	CV1	CV1	CV0	
									Namo
0	0 - 15			vdl	ue				Name



MFX_AVC_WEIGHTOFFSET_STATE

Source: VideoCS

2

Length Bias:

This is a slice level command and can be issued multiple times within a picture that is comprised of multiple slices. The same command is used for AVC encoder (PAK mode) and decoder (VLD and IT modes). However, since for AVC decoder VLD and IT modes, and AVC encoder mode, the implicit weights are computed in hardware, this command is not issued. For encoder, regardless of the type of weight calculation is active for the current slice (default, implicit or explicit), they are all sent to the PAK as if they were all in explicit mode. However, for implicit weight and offset, each entry contains only a 16-bit weight and no offset (offset = 0 always in implicit mode and can be hard-coded inside the hardware). The weights (and offsets) are needed in processing both P and B slice in AVC codec. For P-MB, at most only L0 list is used; for B-MB both L0 and L1 lists may be needed. For a B-MB that is coded in L1-only Prediction, only L1 list is sent.The content of this command matches with the DXVA2 AVC API data structure for explicit prediction mode only: Weights[2][32][3][2] (L0:L1, 0:31 RefPic, Y:Cb:Cr, W:0)

DWord	Bit	Description						
0	31:29	Command Type						
	Default Value:			3h PARALLEL_VIDEO_PIPE				
		Format:		ОрСо	de			
	28:27	Pipeline						
		Default Value:	2h MFX	(_ AVC	_ WEIGHTOFFSET	_STATE		
		Format:	OpCod	e				
	26:24	Media Command Opcode						
		Default Value:			1h AVC_COMM	N		
		Format:			OpCode			
	23:21	SubOpcode A						
		Default Value:				0h		
		Format:				OpCode		
	20:16	SubOpcode B						
		Default Value:				5h		
		Format:				OpCode		
	15:12	Reserved						
		Format:			MBZ			
	11:0	DWord Length						
		Default Value:		60h E:	50h Excludes DWord (0,1)			
		Format: =n To			n Total Length - 2			
1	31:1	Reserved						
		Format:				MBZ		
	0	Weight and Offset Select It must be set in consistent with the WeightedPredFlag and WeightedBiPredIdc in the Img_S command. This parameter is not present in the DXVA.						



			MFX_AVC_W	VEIGHTOFFSET_STATE
		For im	plicit even though only c	one entry may be used, still loading the whole 32-entry table.
		Value	Name	Description
		0	Weight and Offset L0 table	The list that followed is associated with the weight and offset for RefPicList L0
		1	Weight and Offset L1 table	The list that followed is associated with the weight and offset for RefPicList L1
297	31:0	Weight	tOffset	
		Weigh Weigł Weigł Weigł Weigł Weigł Weigł	tOffset[L=L0=0 or L1=1] ntOffset[L][i=0][Y=0][We ntOffset[L][i=0][Cb=1][W ntOffset[L][i=0][Cr=2][W ntOffset[L][i=31][Y=0][W ntOffset[L][i=31][Cb=1][ntOffset[L][i=31][Cr=2][W	<pre>[i=0 to 31][Y=0/Cb=1/Cr=2][weight=0/offset=1] eight=0], WeightOffset[L][i=0][Y=0][Offset=1] Veight=0], WeightOffset[L][i=0][Cb=1][Offset=1] /eight=0], WeightOffset[L][i=0][Cr=2][Offset=1]: /eight=0], WeightOffset[L][i=31][Y=0][Offset=1] Weight=0], WeightOffset[L][i=31][Cb=1][Offset=1] Neight=0], WeightOffset[L][i=31][Cr=2][Offset=1]</pre>
		Forma from - Forma	t for explicit: Both Weigh 128 to 128 at for implicit: S15	nt and Offset are S15 in two's compliment, with a valid range
		This se for eac corres	et of fields is always prese th reference picture, is al conding weight and offs	ent whenever this command is issued. The full table, one entry ways specified. Any reference list L0/L1[i] that does not exist, the let are set to 0.
		Weigł the LO	nt and Offset are 2 byte e WER word and Offset in	each. Apair of Weight and Offset forms a dword, with Weight in the HIGHER word.
		Weigh offset (one-to equal luma_v 2luma_ syntax	ntOffset[L0=0][i=0 to 31] factors applied to the lur p-one correspondence ir to 1, the value of luma_w weight_l0_flag is equal to _log2_weight_denom for element.	[Y=0] (i.e. luma_weight_l0[i]) are specified for the weighting and ma prediction value for list 0 prediction using RefPicList0[i] n i). When luma_weight_l0_flag (Slice Header syntax element) is weight_l0[i] shall be in the range of -128 to 127. When 0 0, luma_weight_l0[i] shall be inferred to be equal to • RefPicList0[i]. luma_log2_weight_denom is a Slice Header
		Weigh weight using I Heade of -128 inferre chrom	ntOffset[L0=0][i=0 to 31] ing and offset factors ap RefPicList0[i] (one-to-or r syntax element) is equa 8 to 127. When chroma_ d to be equal to 2chrom a_log2_weight_denom is	[[Cb=1] (i.e. chromaCb_weight_l0[i]) are specified for the oplied to the chroma Cb prediction values for list 0 prediction ne correspondence in i). When chroma_weight_l0_flag (Slice al to 1, the value of chromaCb_weight_l0[i] shall be in the range weight_l0_flag is equal to 0, chromaCb_weight_l0[i] shall be na_log2_weight_denom for RefPicList0[i]. a Slice Header syntax element.
		Weigh weight using I Heade of -128 inferre	ntOffset[L0=0][i=0 to 31] ing and offset factors ap RefPicList0[i] (one-to-or r syntax element) is equa 8 to 127. When chroma_v d to be equal to 2chrom	[[Cr=2] (i.e. chromaCr_weight_l0[i]) are specified for the oplied to the chroma Cr prediction values for list 0 prediction ne correspondence in i). When chroma_weight_l0_flag (Slice al to 1, the value of chromaCr_weight_l0[i] shall be in the range weight_l0_flag is equal to 0, chromaCr_weight_l0[i] shall be na_log2_weight_denom for RefPicList0[i].



MFX_BSP_BUF_BASE_ADDR_STATE

Source: VideoCS

2

Length Bias:

This frame-level state command is used to specify all the buffer base addresses needed for the operation of the AVC Bit Stream Processing Units (for decoder, it is BSD Unit; for encoder, it is BSE Unit)For both encoder and decoder, currently it is assumed that all codec standards can share the same BSP_BUF_BASE_STATE. The simplicity of this command is the result of moving all the direct MV related processing into the ENC Subsystem. Since all implicit weight calculations and directMV calculations are done in ENC and all picture buffer management are done in the Host, there is no need to provide POC (POC List - FieldOrderCntList, CurrPic POC - CurrFieldOrderCnt) information to PAK. For decoder, all the direct mode information are sent in a separate slice-level command (AVC_DIRECTMODE_STATE command). In addition, in Encoder, the row stores for CABAC encoding and MB Parameters Construction (MPC) are combined into one single row store. The row stores specified in this command do not combine with those specified in the MFC_PIPE_BUF_ADDR_STATE command for hardware simplification reason.

DWord	Bit	Description							
0	31:29	Command Type	Command Type						
		Default Value:		3h PARALLEL_VI	DEO_	PIPE			
		Format:		OpCode					
	28:27	Pipeline							
		Default Value:			2h	Pipeline			
		Format:			Op	oCode			
	26:24	Media Command Opcode							
		Default Value:			0h Co	ommon			
		Format:			ОрСо	ode			
	23:21 SubOpcode A								
		Default Value:	0h MFX	_BSP_BUF_BASE	_ADD	DR_STATE			
	Format: OpCode				2				
	20:16	SubOpcode B							
		Default Value:	4h MFX	<_BSP_BUF_BASE_ADDR_STATE					
		Format:	OpCod						
	15:12	Reserved							
		Format:		MBZ					
	11:0	DWord Length							
		Default Value:		2h Excludes DWord (0,1)					
		Format:		=n Total Lengt	h - 2				
1	31:6	BSD/MPC Row Store Scrat	ch Buff	er Base Addres	s - Re	ead/Write			
		This field provides the ba	ase add	ress of the scra	atch ł	buffer used by BSD (decoder) and			
		MPC (encoder) unit to st	ore MB	information of	f the	previous row for coding each			
		MPC (appendix) bardware	It row.	it is a private bi	utter	used by the BSD (decoder) and			
		buffer must be 64-byte o	achelir	e aligned. Hard	dware	e uses the horizontal address of the			
				5					



MFX_AVC_WEIGHTOFFSET_STATE

current macroblock to address this Row Store.

For AVC BSD, 2 cacheline (CL) per MB when in MBAFF mode (row of MB pair); 1 CL per MB for non-MBAFF. So, to support 256 MBs per row (4K screen resolution), 2 * 256 * 64 bytes = 32,768 bytes are required. Cacheline alignment should be followed. For AVC MPC, 1 cachline for non-MBAFF, 2 cachelines for MBAFF per MB. For VC1, the BSD row store is 512-bit (one cacheline) per MB, times the number of MBs per picture MB row.

BSP Row Store Scratch Buffer - Arbitration Priority Control 5:4 Format: U2 Enumerated Type This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. Value Name 00b **Highest priority** 01b Second Highest priority 10b Third Highest Priority 11b Lowest Priority 30:0 Reserved Format: MBZ 2 31:6 | MPR Row Store Scratch Buffer Base Address - Read/Write (Decoder Only) This field provides the base address of the scratch buffer used by decoder's MPR unit to store MB information of the previous row for decoding each macroblock in the current row. It is a private buffer used by the MPR hardware only. Its content is not accessible by software. **Programming Notes** The MPR Row Store buffer must be 64-byte cacheline aligned. Hardware uses the horizontal address of each macroblock to address the MPR Row Store. Except ILDB Control Data, all other operations does not cross slice boundary. This field is specified in frame-level.2 cacheline (CL) per MB when in MBAFF mode (row of MB pair); 1 CL per MB for non-MBAFF, So, to support 256 MBs per row (4K screen resolution), 2 * 256 * 64 bytes = 32,768 bytes are required. Cacheline alignment should be followed. This field is only valid for AVC decoder mode **MPR Row Store Scratch Buffer - Arbitration Priority Control** 5:4 Format: U2 Enumerated type This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.

		Value	Name	Description					
		0h	[Default]						
		00b	Highest priority	Desc					
		01b	Second highest priority	Desc					
		10b	Third highest priority						
		11b	Lowest priority						
	30:0	Reserved	leserved						
		Format:		MBZ					
3	31:6	Bitplane Rea	d Buffer Base Address						



MFX_AVC_WEIGHTOFFSET_STATE

	It must be cacheline aligned (i.e. 64 bytes address boundary), so lower bit 0 to 5 are used for controlling information. Bitplane buffer is a linear buffer. In VC1 Long format, it is written by application. In VC1 Short Format, it is written and read by H/W only. For VC1 Long Format: it read-only buffer. For VC1 DXVA2 Short Format: it is a write and a read bufferThis field is only valid for VC1 decoder mode.								
5:4	Bitplane Rea	Bitplane Read Buffer - Arbitration Priority Control							
	Format:		U2 Enumerated typ	2 Enumerated type					
	This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.								
	Value		Name		Description				
	00b	Highest priorit	у		Desc				
	01b	Second highes	st priority		Desc				
	10b	Third highest p	oriority						
	11b	Lowest priority							
30:0	Reserved								
	Format:				MBZ				



			MFX_D	BK_OBJ	E	СТ		
Source:		VideoCS						
Length B	Bias:	2						
DWord	Bit			Descript	ior	1		
0	31:29	Command Type						
		Default Value:		3h PARALLEL_V	IDE	O_PIPE		
		Format:		OpCode				
	28:27	Pipeline						
		Default Value:		2h MFX_DB	К_С	OBJECT		
		Format:		OpCode				
	26:24	Media Command Ope	code					
		Default Value:			0h	I Common		
		Format:			Op	oCode		
	23:21	SubOpcode A						
		Default Value:				0h MEDIA_		
		Format:				OpCode		
	20:16	SubOpcode B						
		Default Value:				9h MEDIA_		
		Format:				OpCode		
	15:12	Reserved						
		Format:				MBZ		
	11:0	DWord Length						
		Default Value:		3h Excludes D\	Wo	rd (0,1)		
		Format:		=n				
		Note: Regardless of th	e mode, inline	e data must be	pre	sent in this command		
1	31:6	Pre Deblocking Source	e Address					
		Format:	Graphics	Address[31:6]				
		Specifies the 4K byte aligned frame buffer address for outputting the non-filtered reconstruct YUV picture (i.e. output of final adder in each codec standard, and prior to the deblocking filte unit).						
	5:4	Pre Deblocking - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.						
	Value Name							
		00b Highest priority						
01b Second highest priority								
		10b	Third highest	t priority				
		11b	Lowest priori	ity				
	3	Reserved						



				MFX_DBK_OBJECT					
	2	Pre Deblocking - Graphics Data Type (GFDT) 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	Pre Deblocki	ng - Cac	heability Control					
		This field controls cacheability.							
		Value		Name					
		000	use cacr	headility control bits from GTT entry					
		116	data is r	ached					
2	21.0								
2	31:6	Deblocking C	ontrol A	GraphicsAddrocs[21:6]					
		Specifies the	1K hvte a	ligned frame buffer address as input MB-level deblocking parameters to					
		control the wa each Macrobl	ay hardw ock in ra	are deblock the each micro-block. One 512-bit cacheline is allocated for ster scan order.					
	5:4	Deblocking c	ontrol -	Arbitration Priority Control					
		This field cont	trols the	priority of arbitration used in the GAC/GAM pipeline for this surface.					
		Value	9	Name					
		00b		Highest priority					
		01b		Second highest priority					
		10b		Third highest priority					
		11b		Lowest priority					
	3	Reserved							
	2	Deblocking c This field cont GTT. The effect is ignored for	ontrol - tains the ctive GFD reads.	Graphics Data Type (GFDT) GFDT bit for this surface when writes occur. GFDT can also be set by the T is the logical OR of this field with the GFDT from the GTT entry. This field					
	1:0	Deblocking c This field cont	ontrol - trols cach	Cacheability Control eability.					
		Value		Name					
		00b	use cach	neability control bits from GTT entry					
		01b	data is r	ot cached					
	11b data is cached								
3	31:6	Deblocking [Destinati	on Address					
		Format:		GraphicsAddress[31:6]					
		Specifies the reconstructed	4K byte a YUV pic	ligned frame buffer address for outputting the post-loop filtered ture (i.e. output of the deblocking filter unit)					
	5:4	Deblocking - This field cont	Arbitrat	ion Priority Control priority of arbitration used in the GAC/GAM pipeline for this surface.					



				MFX_DBK_OBJECT						
		Value	e	Name						
		00b		Highest priority						
		01b		Second highest priority						
		10b		Third highest priority						
		11b		Lowest priority						
	3	Reserved								
	2	Deblocking - This field con GTT. The effer is ignored for	Graphic tains the ctive GFD reads.	Is Data Type (GFDT) GFDT bit for this surface when writes occur. GFDT can also be set by the DT is the logical OR of this field with the GFDT from the GTT entry. This field						
	1:0	Deblocking - This field con	Cacheal	bility Control neability.						
		Value		Name						
		00b	use cacl	neability control bits from GTT entry						
		01b	data is r	not cached						
		11b	data is o	cached						
4	31:6	Deblock Row	/ Store A	ddress						
		Format:		GraphicsAddress[31:6]						
		This field prov filter unit to s current row. 1 uses the horiz Store.	vides the tore MB The Deblo contal add	base address of the scratch buffer (read and write) used by the deblocking information of the previous row for filtering of each macroblock in the ocking Filter Row Store buffer must be 64-byte cacheline aligned. Hardware dress of the current macroblock to address the Deblocking Filter Row						
	5:4	Deblock Row Store - Arbitration Priority Control								
		This field con	trols the	priority of arbitration used in the GAC/GAM pipeline for this surface.						
		Value	9	Name						
		00b		Highest priority						
		01b		Second highest priority						
	10b I hird highest priority									
		116		Lowest priority						
	3	Reserved								
	2	Deblock Row This field con GTT. The effer is ignored for	Deblock Row Store- Graphics Data Type (GFDT) 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							



MFX_DBK_OBJECT										
	1:0 Deblock Row Store - Cacheability Control									
		This field controls cacheability.								
	Value Name									
	00b use cacheability control bits from GTT entry		use cacheability control bits from GTT entry							
	01b data is not cached									
		11b	data is cached							



MFX_FQM_STATE

Source: VideoCS

Length Bias:

This is a common state command for AVC encoder modes. For encoder, it represents both the forward QM matrices as well as the decoding QM matrices. This is a Frame-level state. Only Scaling Lists specified by an application are being sent to the hardware. The driver is responsible for determining the final set of scaling lists to be used for decoding the current slice, based on the AVC Spec Table 7-2 (Fall-Back Rules A and B). In MFX AVC PAK mode, PAK needs both forward Q scaling lists and IQ scaling lists. The IQ scaling lists are sent as in MFD in raster scan order. But the Forward Q scaling lists are sent in column-wise raster order (column-by-column) to simplify the H/W. Driver will perform all the scan order conversion for both ForwardQ and IQ.

DWord	Bit		Description						
0	31:29	Command T	Гуре						
		Default Valu	re:		3h P	'A	RALLEL_VIDEO_P	PIPE	
		Format:							
	28:27	Pipeline							
		Default Value: 2h MFX_MULTI_C					WG		
		Format:				(OpCode		
	26:24	Media Com	Media Command Opcode						
		Default Value: 0h MFX_COMMON_STATE					ATE		
		Format:			OpCo	oc	de		
	23:21	SubOpcode	SubOpcode A						
		Default Valu	le:					0h	
		Format:						OpCode	
	20:16 SubOpcode B								
		Default Valu	le:					8h	
		Format:						OpCode	
	15:12	Reserved							
		Format:						MBZ	
	11:0	DWord Len	gth		-				
		Default Valu	le:		20h /	Ex	xcludes DWord ((0,1)	
		Format:			=n T	Го	tal Length - 2		
1	31:2	Reserved							
		Format:						MBZ	
	1:0	MPEG2							
Exists If: //MPEG2- Decoder Only									
	For MPEG2 QM Type: This field specifies which Quantizer Matrix is lo						er Matrix is loaded.		
Value Name									
0 MPEG_INTRA_QUANTIZER_MATRIX									
		1	MPEG_NO	N_INTRA	4_QUA		NTIZER_MATRIX		
2-3 Reserved									



MFX_FQM_STATE									
	1:0	AVC							
		Exists If	:	//AVC- Decoder Only					
		For AVC	QM Type: This fie	eld specifies which Quantizer N	1atrix is loaded.				
		Value		Name					
		0 AVC_4x4_Intra_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)							
		1	1 AVC_4x4_Inter_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)						
		2	2 AVC_8x8_Intra_MATRIX						
		3	AVC_8x8_Inter_M	ATRIX					
233	31:0	Forward	Forward Quantizer Matrix						
		Format:			U32				
		The form unsigned	The format of a Quantizer Matrix is an 8x8 matrix in raster order. Each element is an unsigned byte.						



MFX_IND_OBJ_BASE_ADDR_STATE

Source: VideoCS

Length Bias:

This state command provides the memory base addresses for all row stores, StreamOut buffer and reconstructed picture output buffers required by the MFD or MFC Engine (that are in addition to the row stores of the Bit Stream Decoding/Encoding Unit (BSD/BSE) and the reference picture buffers). This is a picture level state command and is common among all codec standards and for both encoder and decoder operating modes. However, some fields may only applicable to a specific codec standard. All Pixel Surfaces (original, reference frame and reconstructed frame) in the Encoder are programmed with the same surface state (NV12 and TileY format), except each has its own frame buffer base address. In the tile format, there is no need to provide buffer offset for each slice; since from each MB address, the hardware can calculated the corresponding memory location within the frame buffer directly.

The MFX_IND_OBJ_BASE_ADDR command sets the memory base address pointers for the corresponding Indirect Object Data Start Addresses (Offsets) specified in each OBJECT commands. The characteristic of these indirect object data is their variable size (per MB or per Slice). Hence, each OBJECT command must specify the indirect object data offset from the base address to start fetching or writing object data.

While the use of base address is unconditional, the indirection can be effectively disabled by setting the base address to zero. For decoder, there are only 1 read-only per-slice indirect object in the BSD_OBJECT Command, and 2 read-only per-MB indirect objects in the IT_OBJECT CommandFor decoder: the Video Command Streamer (VCS) will perform the memory access bound check automatically using the corresponding MFC Indirect Object Access Upper Bound specification. If any access is at or beyond the upper bound, zero value is returned. The request to memory is still being sent, but the corresponding codec's BSD unit will detect this condition and perform the zeroing return. If the Upper Bound is turned off, the beyond bound request will return whatever on the bus (invalid data). For encoder, there are 1 read-only per-MB indirect object in the PAK_OBJECT Command, and 1 write-only per-slice indirect object in the PAK Slice_State CommandFor encoder: whenever an out of bound address accessing request is generated, VMX will detect such requests and snap the address to the corresponding [indirect object base address + indirect data start address]. VMX will return all 0s as the data to the requester. NotationDefinitionPhysicalAddress[n:m] Corresponding bits of a physical graphics memory byte address (mapped by a GTT).

DWord	Bit			Description				
0	31:29	Command Type						
		Default Value:		3h PARALLEL_VIDEO_PIPE				
		Format:		OpCode				
	28:27	Pipeline	Pipeline					
		Default Value:	2h MFX_IND_OBJ_BASE_ADDR_STATE					
		Format:	OpCode					
	26:24	Common Opcode						
		Default Value:	0h MF	K_IND_OBJ_BASE_ADDR_STATE				
		Format:	OpCoc	le				
	23:21	Sub OpcodeA						
		Default Value:	0h MFX_IND_OBJ_BASE_ADDR_STATE					
		Format:	OpCoc	le				
	20:16	SubOpcodeB						



		Ν	IFX_IN	D_0	BJ_BASE_ADDR_STATE					
		Default Va	lue:	3h	MFX_IND_OBJ_BASE_ADDR_STATE					
		Format:		Op	Code					
	15:12	Reserved								
		Format:			MBZ					
	11:0	DWord Length								
		Default Va	lue:		0009h Excludes DWord (0,1)					
		Format:			=n Total Length - 2					
1	31:12	MFX Indire	ect Bitstrea	m Objec	t - Base Address (Decoder and Stitch Modes)					
		Format:		Grap	hicsAddress[31:12]					
		pointed in Data. This f	the MFD_XX ield is only	(X_BSD_(valid in N	nemory base address for the read-only indirect data object OBJECT command for fetching (reading) the compressed Slice MPEG2, AVC and VC1 decoder VLD mode.					
	11:6	Reserved								
		Format:			MBZ					
	5:4	MFX Indir	ect BSD Ob	ject - Ar	bitration Priority Control					
		Format:		U2	2 Enumerated Type					
		This field co	ontrols the	oriority o	of arbitration used in the GAC/GAM pipeline for this surface.					
		Va	lue		Name					
		000		Highest	priority					
		10b		Second	abost priority					
		100 11b			priority					
	2	Becomicad		LOwest	phonty					
	3	Keserved	at Ditetroa	m Ohia	t Cranhier Data Tuma (CEDT)					
	2	Format:	ect bitstrea	m Objec						
		This field co	ontains the	GFDT bit	for this surface when writes occur. GFDT can also be set by the					
		GTT. The ef	fective GFD	T is the l	ogical OR of this field with the GFDT from the GTT entry. This field					
		is ignored f	for reads.							
	1:0	MFX Indire	ect Bitstrea	m Objec	t - Cacheability Control					
		Format: U2 Enumerated Type								
Inis field controls cacheability.										
00b GTT optry Use cacheability control bits from CTT optry										
		00b 01b	Not cacher	ached data is not cached						
		11b	Cached		data is cached					
2	31.12	MEX India	ect Ritetree	m Obier	t - Access Upper Bound (Decoder and Stitch Modes)					
2	51.12	Format:		Gran	hicsAddress[31:12]					
				0.00						



MFX_IND_OBJ_BASE_ADDR_STATE

		This field s the indirect Indirect da field to 0 v the MFX In is not pres 0.This field	This field specifies the 4K-byte aligned (exclusive) maximum Graphics Memory address access by the indirect data object in the MFD_XXX_BSD_OBJECT command for the compressed Slice Data. Indirect data accessed at this address and beyond will return as 0 by the hardware. Setting this field to 0 will cause this range check to be ignored. If non-zero, this address must be greater than the MFX Indirect Bitstream ObjectBase Address state. Hardware ignores this field if indirect data s not present, i.e. the Indirect Data Length field of the MFD_XXX_BSD_OBJECT command is set to D.This field is only valid in MPEG2, AVC and VC1 decoder VLD mode.						
	11:0	Reserved							
		Format:					MBZ		
3	31:12	MFX Indi	rect MV Obj	ect - Ba	ase A	Address			
		Format:		Gra	phic	sAddress[31:12]			
		pointed in command AVC decor	the encode for fetching der IT mode	r MFC_A the per	merr AVC_ r-MB	PAK_OBJECT command of MV data. This field is or	or the de	ecoder MFD_IT_OBJECT in AVC encoder mode or in	
	11:6	Reserved							
		Format: MBZ							
	5:4	4 MFX Indirect MV Object - Arbitration Priority Control							
		Format:		U	J2 En	umerated Type			
		This field o	controls the	priority	of ar	bitration used in the GA	C/GAM	pipeline for this surface.	
		Va	alue			N	ame		
		00b		Highes	st prie	ority			
		01b		Second	d hig	hest priority			
		10b		Third h	nighe	est priority			
		11b		Lowest	t prio	prity			
	3	Reserved							
	2	MFX Indi	rect MV Obj	ect - Gr	raph	ics Data Type (GFDT)			
		Format:						U1	
		This field of	contains the	GFDT bi	it for	this surface when writes	S OCCUR.	GFDT can also be set by the	
		is ignored	for reads.		logi			r nom the Grr entry. This held	
	1:0	MFX Indi	rect MV Obj	ect - Ca	ache	ability Control			
		Format: U2						U2	
		This field controls cacheability.							
		Value	Na	me			Descri	ption	
		00b	From GTT e	entry		use cacheability control	bits fro	m GTT entry	
		01b	Not cached			data is not cached			
		11b	Cached			data is cached			



		N	IFX_IN	D_OB.	J_BASE_ADD	R_STATE				
4	31:12	MFX Indir	ect MV Obj	ect Access	Upper Bound					
		Format:		Graphic	sAddress[31:12]					
		This field s the indirec MB MV da Setting thi greater tha indirect da MFD_IT_O decoder IT	rect data object in the MFC_AVC_PAK_OBJECT / MFD_IT_OBJECT command for the per- data. Indirect data accessed at this address and beyond will return as 0 by the hardware. this field to 0 will cause this range check to be ignored. If non-zero, this address must be than the MFX Indirect MV Object Base Address state. Hardware ignores this field if data is not present, i.e. the Indirect Data Length field of the MFC_AVC_PAK_OBJECT / _OBJECT command is set to 0.This field is only valid in AVC encoder mode or in AVC r IT mode.							
	11:0	Reserved								
		Format:				MBZ				
5	31:12	MFD Indi	rect IT-COE	FF Object -	Base Address (Decode	r Only)				
		Format:		Graphic	sAddress[31:12]					
Specifies the 4K-byte aligned memory base address for the read-only indirect dat pointed in the MFD_IT_OBJECT command for fetching (reading) the per-MB non-s coefficient data (all inverse scaling and quantization are done in hardware). This fi in MPEG2, AVC and VC1 decoder IT mode.						e read-only indirect data object ding) the per-MB non-scaled one in hardware). This field is only valid				
	11:6	Reserved								
		Format:				MBZ				
	5:4	MFD Indi	rect IT-COE	FF Object -	Arbitration Priority Co	ntrol				
		Format:		U2 Er	numerated Type					
		This field o	controls the	priority of a	rbitration used in the GA	C/GAM pipeline for this surface.				
		Va	alue		Ν	lame				
		00b		Highest pri	ority					
		01b		Second hig	hest priority					
		10b		Third highe	est priority					
		11b		Lowest pric	ority					
	3	Reserved								
	2	MFD Indi	rect IT-COE	FF Object -	Graphics Data Type (G	FDT)				
		Format:		050711.C		U1				
		This field contains the GFDT bit for this surface when writes occur. GFDT can also be set by the								
		is ignored	for reads.	i is the logi		ale of britten ale of renay. This held				
	1.0	MFD Indi	rect IT-COFI	FF Object -	Cacheability Control					
	1.0	Format:			numerated type					
		This field o	controls cach	neability.						
		Value	Na	me		Description				
		00b	From GTT e	entry	use cacheability contro	l bits from GTT entry				



MFX_IND_OBJ_BASE_ADDR_STATE

		01b	Not cached		data is not cached				
		11b	Cached		data is cached				
6	31:12	MFD Indi	rect IT-COEFF O	bject -	Access Upper Bound (I	Decoder Only)			
		Format:		Graphie	csAddress[31:12]				
		This field	specifies the 4K-b	oyte ali	gned (exclusive) maximu	m Graphics Memory address	access by		
		the indire	ct data object in t	the MF	D_IT_OBJECT command f	for the per-MB non-scaled co	pefficient		
		this field t	o 0 will cause this	s range	e check to be janored. If i	non-zero, this address must	e. Setting be greater		
		than the N	/IFD Indirect IT-C	OEFF C	bject Base Address state	e. Hardware ignores this field	if indirect		
		data is not present, i.e. the Indirect COEFF Data Length field of the MFD_IT_OBJECT command is							
		set to 0. This held is only valid in MPEG2, AVC and VCI decoder 11 mode.							
	11:0	Reserved							
		Format:				MBZ			
7	31:12	MFD Indi	rect IT-DBLK Ob	ject - I	Base Address (Decoder	Only)	J		
		Format:		. Graphio	csAddress[31:12]				
		Specifies t	he 4K-byte align	ed mer	mory base address for th	e read-only indirect data obj	ect		
		pointed in	the MFD_IT_OBJ	ECT co	ommand for fetching (rea	ding) the per-MB Deblocking	g filter		
		control de		ny van					
	11:6	Reserved							
		Format:				MBZ			
	5:4	MFD Indi	rect IT-DBLK Ob	ject - /	Arbitration Priority Con	itrol			
		Format:		U2 Ei	numerated Type				
		This field	controls the prior	ity of a	rbitration used in the GA	C/GAM pipeline for this surf	ace.		
		V	alue		N	lame			
		00b	Hig	hest pr	iority				
		016	Sec	ond hig	ghest priority				
		10b	Thir	d high	est priority				
		116	Low	est pri	ority				
	3	Reserved							
	2	MFD Indi	rect IT-DBLK Ob	ject - (Graphics Data Type (GF	DT)			
		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	MFD Indi	rect IT-DBLK Ob	ject - (Cacheability Control				
		This field	controls cachaabi	U2 EI	numerated Type				
		Value	Name	inty.	Descr	intion			
		Tanac	itunic		Desci				



		1	MFX_IN	D_C	BJ_BASE_ADD	R_ST	ATE			
		00b	From GTT en	try	use cacheability control bits	s from G	TT entry			
		01b	Not cached		data is not cached			_		
		11b	Cached		data is cached					
8	31:12 MFD Indirect IT-DBLK Obje		Objec	Access Upper Bound (Decoder Only)						
		Format:		Gra	aphicsAddress[31:12]					
		Format:		Gra	aphicsAddress[31:12]					
		This field the indire data. Ind this field than the data is no comman	rect data object in the MFD_IT_OBJECT command for the per-MB Deblocking filter control direct data accessed at this address and beyond will return as 0 by the hardware. Setting d to 0 will cause this range check to be ignored. If non-zero, this address must be greater MFD Indirect IT-DBLK Object Base Address state. Hardware ignores this field if indirect not present, i.e. the Indirect Deblocking Control Data Length field of the MFD_IT_OBJECT and is set to 0.This field is only valid in AVC decoder IT mode.							
	11:0	Reserved	d							
		Format:	-			MBZ				
9	31:12	MFC Ind	irect PAK-BS	E Obie	ct - Base Address (Encoder	Only)				
5	51.12	Format:		Gra	aphicsAddress[31:12]	<u>e</u> j)				
		Specifies	the 4K-byte a	ligned	memory base address for the	e write-o	only indirect data	object		
		pointed in the PAK_SLICE_STATE command for writing out the compressed bitstream. This field only valid in AVC encoder mode.								
	11:6	Reserved	d							
		Format:				MBZ				
	5:4	MFC Ind	irect PAK-BS	E Obje	ct - Arbitration Priority Cor	ntrol				
		Format:		(J2 Enumerated Type					
		This field	controls the p	oriority	of arbitration used in the GA	AC/GAM	pipeline for this s	urface.		
			/alue		Ν	lame				
		00b		Highes	st priority					
		01b		Secon	d highest priority					
		10b		Third h	nighest priority					
		11b		Lowest	t priority					
	3	Reserve	b							
	2	MFC Ind	irect PAK-BS	E Obje	ct - Graphics Data Type (GF	DT)	-			
		Format:					U1			
		This field	contains the	GFDT b	it for this surface when write	s occur.	GFDT can also be	set by the		
		GTT. The is ignore	effective GFD d for reads.	T is the	e logical OR of this field with	the GFD	T from the GTT er	try. This field		
	1:0	MFC Ind	irect PAK-BS	E Obje	ct - Cacheability Control					
		Format:		l	J2 Enumerated Type					



MFX IND	OBJ	BASE	ADDR	STATE
	_		-	

		This field c	his field controls cacheability.							
		Value	Name		Description					
		00b	GTT entry	use cacheability control bits	s from GTT entry					
		01b	Not cached	data is not cached						
		11b	Cached	data is cached						
10	31:12	MFC Indire	MFC Indirect PAK-BSE Object - Access Upper Bound (Encoder Only)							
		Format:	Format: GraphicsAddress[31:12]							
		This field specifies the 4K-byte aligned (exclusive) maximum Graphics Memory address access by the indirect data object in the PAK_SLICE_STATE command for the per-slice output bitstream. Indirect data accessed at this address and beyond will be blocked by the hardware and ignored. Setting this field to 0 will cause this range check to be ignoredIf non-zero, this address must be greater than the MFC Indirect PAK-BSE Object Base Address state. This field is only valid in AVC encoder mode.								
	11:0	Reserved								
		Format:			MBZ					



MFX_JPEG_HUFF_TABLE_STATE

Source: VideoCS

2

Length Bias:

This Huffman table commands contains both DC and AC tables for either luma or chroma. Once a Huffman table has been defined for a particular destination, it replaces the previous tables stored in that destination and shall be used in the remaining Scans of the current image. A Huffman table will be sent to H/W only when it is loaded from bitstream.

DWord	Bit	Description							
0	31:29	Command Type	Command Type						
		Default Value:		3h P	ARALLEL_VIDEO_F	PIPE			
		Format:		OpC	Code				
	28:27	Pipeline			_				
		Default Value:			2h MFX_MULTI_[DW			
		Format:			OpCode				
	26:24	Media Command	Opcode						
		Default Value:			7h JPEG_COMM	ON			
		Format:			OpCode				
	23:21	SubOpcode A							
		Default Value:				0h			
		Format:				OpCode			
	20:16	SubOpcode B							
		Default Value:				2h			
		Format:				OpCode			
	15:12	Reserved							
		Format:				MBZ			
	11:0	DWord Length							
		Default Value:	(033Dh	Excludes DWord	(0,1)			
		Format:	=	=n Tot	tal Length - 2				
1	31:1	Reserved							
		Format:				MBZ			
	0	HuffTableID (1-bi	t)						
		Identifies the huffm	han table.	Description					
		Value	v		Huffman table fo	Description			
2.4	21.0		Y Huttman table for Y						
24	31:0	DC_BITS (12 8-bit array) The number of DC Huffman codes of length i, where i is 1~12							
57	31:0	DC_HUFFVAL (12 8-bit array)							
		The value associate	ed with each D	C Huf	fman code of leng	ıth i.			
811	31:0	AC_BITS (16 8-bit the list of Li, numbe	array) er of Huffman	codes	s of length i, where	e i is 1~16			



MFX_JPEG_HUFF_TABLE_STATE								
1251	31:0	AC_HUFFVAL (160 8-bit array)						
		the list of vij, the value associated with each Huffman code of length i						
52	31:16	Reserved						
		Format:	MBZ					
	15:0	AC_HUFFVAL(2-8 bit array) In AC table, BITS can have up to 16-bit codeword. Li car of likely random distributed values	n be 0 ~ 162. HUFFVAL will have a list					



			MFX_JF	PEG_P	IC_S	T	ATE					
Source:		VideoCS										
Length E	Bias:	2										
DWord	Bit			D	escripti	ion						
0	31:29	Command Type										
		Default Value:		3h PARA	LLEL_VI	DE	O_PIPE					
		Format:		OpCode								
	28:27	Pipeline										
		Default Value:		2h	MFX_M	1UL	.TI_DW					
		Format:		Op	Code							
	26:24	Media Command	l Opcode									
		Default Value:					7h JPEG					
_		Format:					OpCode					
	23:21	SubOpcode A										
		Default Value:				0h	Common					
		Format:				Ор	Code					
	20:16	SubOpcode B										
		Default Value:				0h MEDIA_						
		Format:		OpCode								
	15:12	Reserved										
		Format: MBZ										
	11:0	DWord Length										
		Format:	=n 1	Total Leng	th - 2							
		Value	Name	2			Descript	tion				
		0001h	[Default]		Exclud	es l	DWord (0,1)					
1	31:21	Reserved			1							
		Exists If:		//Decode	er Only							
		Format:		MBZ								
	20:19	Reserved										
		Exists If:		//Decode	er Only							
		Format:	MBZ									
	18	Reserved										
		Exists If:		//Decode	er Only							
		Format:		MBZ	MBZ							
	17:16	Reserved										
		Exists If:		//Decode	er Only							
		Format:		MBZ								



			MFX_JP	PEG_PIC_STATE				
15:12	Reserve	ed						
	Exists I	f:		//Decoder Only				
	Format	t:		MBZ				
11:8	Reserve	Reserved						
	Exists I	f:		//Decoder Only				
	Format	t:		MBZ				
7:6	Reserve	ed						
	Exists I	f:		//Decoder Only				
	Format	t:		MBZ				
5:4	Rotatic	on		<u></u>				
5.1	Exists I	f:		//Decoder Only				
	Value	Name		Description				
	00b		no rotation	protation				
	01b		rotate clockwise S	ate clockwise 90 degree				
	10b		rotate counter-clo	counter-clockwise 90 degree (same as rotating 270 degree clockwise)				
	11b		rotate 180 degree	ee (NOT the same as flipped on the x-axis)				
3	Reserve	ed						
	Exists I	f:		//Decoder Only				
	Format	t:		MBZ				
2:0	Input F	ormat Y	UV					
	Exists I	f:		//Decoder Only				
	Format	t:		U3				
	Value	Name		Description				
	0	Default	YUV400 (grays	ayscale image)				
	1	[Berau.	VI IV420					
	2		VIIV422H 2Y (Horizontally chroma 2.1 subsampled) - horizontal 2 Y-block				
	2		1U and 1V					
	3		YUV444					
	4		YUV411					
	5		YUV422V_2Y (* 1V	Vertically chroma 2:1 subsampled) - vertical 2 Y-blocks, 1U and				
	6		YUV422H_4Y -	Y - 2x2 Y-blocks, vertical 2U and 2V				
	7		YUV422V_4Y -	- 2x2 Y-blocks, horizontal 2U and 2V				
31:30		ed						
0 0	Exists I	f:		//Decoder Only				
	Format	t:		MBZ				



29	Reserved							
	Exists If:	//Decoder Only						
	Format:	MBZ						
28:16	Frame Height In Blo	cks Minus 1						
	Exists If:	//Decoder Only						
	Format:	U13-1						
15.12	For interleaved components, $((Y + (V_1^* - L)) / (V_1^* - L)) - L$, where "/" is integer division. For non-interleaved components, $((Y + 7) / 8) - 1$.							
19.19	Exists If:	//Decoder Only						
	Format:	MBZ						
12:0	Frame Width In Blocks Minus 1							
	Exists If:	//Decoder Only						
	Format:	U13-1						
		(The number of blocks in width) - 1. This value is calculated using the number of samples per line X and horizontal sampling factor the first component H ₁ in Frame header. See the note following this table. For interleaved components, (((X + (H ₁ *8 -1)) / (H ₁ *8)) * H ₁) - 1. For non-interleaved components, ((X + 7) / 8) - 1.						



MFX_MPEG2_PIC_STATE

Source: VideoCS

Length Bias:

This must be the very first command to issue after the surface state, the pipe select and base address setting commands. For MPEG-2 the encoder is called per slice-group, however the picture state is called per picture. Notice that a slice-group is a group of consecutive slices that no non-trivial slice headers are inserted in between.

DWord	Bit				Description			
0	31:29	Command	Command Type					
		Default Va	alue:	3h	3h PARALLEL_VIDEO_PIPE			
		Format:		0	oCode			
	28:27	Pipeline						
		Default Va	alue:	2h	MFX_MPEG2_PIC_S	STATE		
		Format:		Ор	Code			
	26:24	Media Co	mmand Opcode					
		Default Va	alue:		3h MPEG2_COMM	ION		
		Format:			OpCode			
	23:21	SubOpcoc	le A					
		Default Va	alue:		0h			
		Format:		OpCode				
	20:16	SubOpcode B						
		Default Va	alue:		0h			
		Format:				OpCode		
	15:12	Reserved						
		Format:	MBZ					
	11:0	DWord Length						
		Default 0h Excludes DWord (0,1)= 00Bh, used for normal decode and encode						
		Value:	mode000h, a special case to provide a dummy image state for stitch mode					
			command are ignor	and are ignored by hardware.				
		Format: =n Total Length - 2						
1	31:28	f code[1][[1].					
		Used for b	ackward motion vect	or p	orediction. See ISO/	/IEC 13818-2 7.6.3.1 for details		
	27:24	f_code[1][[0].					
		Used for b	ackward motion vect	or p	prediction. See ISO/	/IEC 13818-2 7.6.3.1 for details		
	23:20	f_code[0][Used for fo	[1] prward motion vector	r pre	ediction. See ISO/IE	C 13818-2 7.6.3.1 for details		
	19:16	f_code[0][Used for fo	[0] prward motion vector	r pre	ediction. See ISO/IF	C 13818-2 7.6 3.1 for details		
	15:14	Intra DC P	Precision	Pre				



MFX	MPEG2	PIC	STATE
		_	-

		Format: U2								
		See ISO/IEC 13818-2 6.3.10 for details.								
	13:12	This field specifies whether the picture is encoded in the form of a frame picture or one field (top or bottom) picture. See ISO/IEC 13818-2 6.3.10 for details. Format = MPEG_PICTURE_STRUCTURE00 = Reserved01 = MPEG_TOP_FIELD10 = MPEG_POTTOM_STRUCTURE01 = MPEG_FORMES								
		MPEG_BOTTOM_FIELD11 = MPEG_FRAME								
	11	TFF (Top Field First) When two fields are stored in a picture, this bit indicates if the top field is the first field. For a frame P picture, the value 1 indicates that the top field of the reconstructed frame is the first field output by the decoding process, the same as defined in ISO/IEC 13818-2 6.3.10. Particularly, it is used by the hardware to calculate derivative motion vectors from the dual-prime motion vectors. For a field P picture, hardware uses this bit together with the Picture Structure to determine if the current picture is the Second Field. In this case, the definition of this bit differs from ISO/IEC 13818-2 6.3.10 - software must derive the value for this bit according to the following relation:Picture Structure = top fieldPicture Structure = bottom fieldSecond Field = 0TFF = 1TFF = 0Second Field = 1TFF = 0TFF = 1								
	10	Frame Prediction Frame DCT This field provides constraints on the DCT type and prediction type. It affects the syntax of the bitstream.								
	9	Concealment Motion Vector Flag This field indicates if the concealment motion vectors are coded in intra macroblocks. It affects the syntax of the bitstream.								
	8	Quantizer So	ale Type							
		Format:		MPEG_Q_SCALE_TYPE						
		This field spe	cifies the qua	antizer scaling type.						
		Value	Name	De	scription					
		0h		MPEG_QSCALE_LINEAR						
		1h		D MPEG_QSCALE_NONLINE	EAR esc					
	7	Intra VLC Format This field is used by VLD								
	6	Scan Order								
		Format:	MP	EG_INVERSESCAN_TYPE						
		This field specifies the Inverse Scan method for the DCT-domain coefficients in blocks of the current picture.								
		Value	Name	Des	scription					
		0h		MPEG_ZIGZAG_SCAN						
		1h		MPEG_ALTERNATE_VERTICA	L_SCAN					
	5:0	Reserved								
2	31:24	Reserved								
		Format:			MBZ					



	MF	X_I	MPEG2_PIC_STATE						
23:15	Reserved	eserved							
	Format:		MBZ						
14	LoadSlicePoint	erFla	lag - LoadBitStreamPointerPerSlice						
	Exists If:		//Encoder						
	To support mult an encoded slice first slice of a fra together to form pointer is loaded a frame will be y	o support multiple slice picture and additional header/data insertion before and after n encoded slice. When this field is set to 0, bitstream pointer is only loaded once for th rst slice of a frame. For subsequent slices in the frame, bitstream data are stitched ogether to form a single output data stream. When this field is set to 1, bitstream ointer is loaded for each slice of a frame. Basically bitstream data for different slices of frame will be written to different memory locations.							
	Value Name Description								
	0h	0h Load BitStream Pointer only once for the first slice of a frame							
	1h	1h Load/reload BitStream Pointer only once for the each slice, reload start location of the bitstream buffer from the Indirect PAK-BSE OI Data Start Address field							
13	Reserved								
	Format: MBZ								
	was Concealment Enable								
12	Reserved								
	Format:		MBZ						
	was Concealme	nt Re	eterence						
11	Reserved								
	Format:		MBZ						
	was Concealme	nt Ty	уре						
10:9	Picture Coding	Тур	pe						
	Format:	I	MPEG_PICTURE_CODING_TYPE						
	This field identif picture (P) or bidetails.	ies w dire	whether the picture is an intra-coded picture (I), predictive-coded ectionally predictive-coded picture (B). See ISO/IEC 13818-2 6.3.9 fc						
	Value		Name						
	00b		Reserved						
	01b		MPEG_I_PICTURE						
	10b		10 = MPEG_P_PICTURE						
	11b		MPEG_B_PICTURE						
8:2	Reserved								
	Format:		MBZ						
	was Slice Error C	Contr	trol						



		MF	X_MPI	G2_PIC_STAT	E						
	1	MismatchCont	MismatchControlDisabled								
	These 2 bits flag disables mismatch control of the inverse transformation										
		some specific cases during reference reconstruction.									
		Value Name Description									
		00b Mismatch control applies to all MBs									
		01b	Disable mismatch control to all intra MBs whose all AC-coefficients are zero.								
		10b	Disable mis	match control to all MBs v	whose all AC-coefficients are zero.						
		11b	Disable mis	match control to all MBs.							
	0	Disable Misma	Itch	ad point arithmatic correct	ion						
2	21	Pecorved									
5	21	Format:			MBZ						
	30.20	Reserved									
	50.29	Format:			MBZ						
	28.24	Reserved									
	20.24	Reserved									
	15.8	Reserved									
	15.0	Format:	MBZ for fu	uture supporting width > 4	4K						
	7:0	Reserved									
4	31:16	MinFrameWSi	ze								
		Format:			U16						
		- Minimum Frame Size [15:0] (16-bit) (Encoder Only)Mininum Frame Size is specified t									
		compensate for	Rate Contr	ol Currently zero fill (no ne	eed to perform emulation byte						
		slice of a nictur	ne only to tr e. Encoder n	arameter not part of DXV	2 WORD Insertion (If any) at the last A The caller should always make						
		sure that the va	llue, represe	nted by Mininum Frame S	ize, is always less than maximum						
		frame size Fram	neBitRateMa	x (DWORD 10 bits 29:16).	This field is reserved in Decode						
		mode.		_							
		Value	Name		Description						
		[0,0003FFFFh]		The programmable range	e when MinFrameWSizeUnits is 00.						
	e when MinFrameWSizeUnits is 01.										
		[0,03FFFFFFh]		The Programmable range	e when MinFrameWSizeUnits is 10.						
		[0,FFFFFFFF]		The Programmable range	e when MinFrameWSizeUnits is 11.						
		0h	[Default]								
	15	Reserved									
		Format:			MBZ						
	14:12	RoundInterAC	,								



			MF	X_N	IPEG2_PIC_STATE					
		rounding precision for non-Intra AC000: +1/16001: +2/16010: +3/16011: +4/16100: +5/16101: +6/16110: +7/16111: +8/16								
	11	Reserve	Reserved							
		Format	:		MBZ					
	10:8	RoundI	ntraAC							
		Format	:		U3					
		roundin +5/161	rounding precision for Intra AC000: +1/16001: +2/16010: +3/16011: +4/16100: +5/16101: +6/16110: +7/16111: +8/16							
	7	Reserve	ed							
		Format			MBZ					
	6:4	RoundI	nterDO							
		roundin +5/161	rounding Precision for non-Intra-DC000: +1/16001: +2/16010: +3/16011: +4/16100: +5/16101: +6/16110: +7/16111: +8/16							
	3	Reserve	ed							
		Format	:		MBZ					
	2:1	RoundIntraDC rounding Precision for Intra-DC00: +1/801: +2/810: +3/811: +4/8								
	0	Reserve	ed							
5	31:17	Reserved								
		(for future Mask bits)								
	16	FrameS	ısk							
		Frame s	ize con	forman I	ce maskThis field is used when MacroblockStatEnable is set to 1.					
		Value	Name	_	Description					
		Uh		Do no [.] MFC_N frame	MPEG2_SLICE Quantization Parameter values in MPEG2_SLICEGROUP_STATE with suggested slice QP value for elevel Rate control					
		1h		Replac	ce Slice Quantization Parameter values in					
				MFC_MPEG2_SLICEGROUP_STATE with suggested slice QP value fo						
				frame	level Rate control values in MFC_IMAGE_STATUS control register.					
	15:13	Reserve	ed							
	12	InterMBForceCBPZeroControlMask								
		Format	:		U1					
		Inter M	B Force	CBP ZE	RO mask.					
		Val	ue	Name	Description					
		[0,FFFF	FFFFh]							
		0h			No effect					
		1h			Zero out all A/C coefficients for the inter MB violating Inter Confirmance					
	11:10	MinFra	meWSi	zeUnit						
		This field is the Minimum Frame Size Units								



MFX_MPEG2_PIC_STATE

	Value	1	Name	Description		
	00b	compat mode	tibility	Minimum Frame Size is in old mode (words, 2bytes)		
	01b	16 byte	2	Minimum Frame Size is in 16bytes		
	10b	4Kb		Minimum Frame Size is in 4Kbytes		
	11b 16Kb N			Minimum Frame Size is in 16Kbytes		
9	MBRate MB Rate disabled Buffer.	e Contro e Contro d or MB	olMask ol conforman level Rate cc	nce maskThis field is ignored when MacroblockStatEnal control flag for the current MB is disable in Macroblock	ble is Status	
	Value	Name		Description		
	0h		Do not chan in Macroblo	Jo not change QP values of inter macroblock with suggested QP v n Macroblock Status Buffer		
	1h		Apply RC QF	apply RC QP delta for all macroblock		
8	Reserve	ed				
7	Reserve	Reserved				
	Format	t:	MBZ			
6:4	Reserve	ed				
J	This is a FrameB	itRateM	in.	g if the condition of frame level bit count is less than		
	Value	Name		Description		
	0h	Disable	Do not upo	date bit0 of MFC_IMAGE_STATUS control register.		
	1h	Enable	set bit0 and total frame rate Minim	d bit 1of MFC_IMAGE_STATUS control register if the e level bit counter is less than or equal to Frame Bit um limit.		
2	Frame	BitRate	MaxReportN	1ask 🛛		
	This is a FrameB	a mask b itRateM	oit controlling ax.	g if the condition of frame level bit count exceeds		
	Value	Name		Description		
	0h	Disable	Do not upo	date bit0 of MFC_IMAGE_STATUS control register.		
	1h Enable set bit0 and bit 1 of MFC_IMAGE_STATUS control register if the total frame level bit counter is greater than or equal to Frame Bit rate Maximum limit.			d bit 1 of MFC_IMAGE_STATUS control register if ame level bit counter is greater than or equal to rate Maximum limit.		
1	InterM This is a InterME	BMaxSi a mask b 3MaxSiz	3MaxSizeReportMask mask bit controlling if the condition of any inter MB in the frame exceeds MaxSize.			
	Value	Name		Description		
	0h		Do not upda	ate bit0 of MFC_IMAGE_STATUS control register.		
	1h		set bit0 of N	t bit0 of MFC_IMAGE_STATUS control register if the total bit of		



			MF	X_MPEG2_PIC	C_STATI	E			
				limit.					
	0	IntraM This is a IntraMB	IntraMBMaxSizeReportMask This is a mask bit controlling if the condition of any intra MB in the frame exceeds IntraMBMaxSize						
		Value	Value Name Description						
		0h	0h Do not update bit0 of MFC_IMAGE_STATUS control register.						
		1h		set bit0 of MFC_IMAGE_STATUS control register if the total bit counter for the current MB is greater than the Intra MB Conformance Max size limit.					
6	31:28	Reserve	ed				1		
[Eviate][]En en ele		Format	Format: MBZ						
Only	27:16	InterM	BMaxSi	ize					
- ,		Default	t Value:				FFFh		
		This field, Inter MB Conformance Max size limit, indicates the allowed max bit count size for Inter MB							
	15:12	Reserve	ed						
		Format	t:			MBZ			
	11:0	IntraM	BMaxSi	ize			1		
		Default	t Value:				FFFh		
		This fiel for Intra	d, Intra a MB	MB Conformance Max siz	e limit,indicat	es the allo	owed max bit count size		
7	31:1	Reserve	ed						
	-								
	0	Reserve	ed						
	21.24	Format: MBZ							
8	31:24	SliceDe		lax[3]			57		
[ExistsIf]Encode		FOIMA	Format: 5/						
Only		This fie 1/8 reg MFC_IN FrameE i.e., in t Range:	This field is the Slice level delta QP for total bit-count above FrameBitRateMax - first 1/8 regionThis field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS control register when total bit count for the entire frame exceed FrameBitRateMax but is within 1/8 of FrameBitRateMaxDelta above FrameBitRateMax i.e., in the range of (FrameBitRateMax, (FrameBitRateMax+ FrameBitRateMaxDelta>> Range: [-30,30]						
				Value		N	ame		
		0h			Disable				
		1h			Enable				
	23:16	SliceDe	ltaQPN	lax[2]					



MFX	MPEG2	PIC	STATE

		Format:	S7			
		Range: [-30,30]				
		This field is the Slice level delta QP for bit-count above FrameBitRateMax - above 1/8 and below 1/4 This field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS control register when total bit count for the entire frame is between 1/8 and ¹ / ₄ of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta>>3), (FrameBitRateMax+ FrameBitRateMaxDelta>>2).				
	15:8	SliceDeltaQPMax[1]				
		Format:	S7			
		-				
		Range: [-30,30]				
		This field is the Slice level delta QP for bit-count above FrameBitRateMax - above1/4				
		MFC_IMAGE_STATUS control register when total bit count for	the entire frame is			
		between $\frac{1}{4}$ and $\frac{1}{2}$ of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range				
		of ((FrameBitRateMax+ FrameBitRateMaxDelta>>2), (FrameBitRateMaxDelta>>1)	itRateMax+			
	7.0					
	7.0	Format:	57			
		Range: [-30,30]				
		This field is the Slice level delta QP for bit-count above FrameBitRateMax - above 1/ 2This field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS control register when total bit count for the entire frame is above FrameBitRateMax more than half the distance of FrameBitRateMaxDelta , i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta>>1), infinite).				
9	31:24	SliceDeltaQPMin[3]				
[ExistsIf]Encode Onlv		Format:	S7			
		Bange: [20 20]				
,		This field is the Slice level delta OP for total bit-count below F	rameBitBateMin - first			
		1/8 regionThis field is used to calculate the suggested slice QP into the MFC IMAGE STATUS control register when total bit count for the entire frame is less				
		than FrameBitRateMin and greater than or equal to 1/8 the distance of				
		FrameBitRateMinDelta from FrameBitRateMin, i.e., in the range of [(FrameBitRateMin- FrameBitRateMinDelta>>3), FrameBitRateMin).				
	23:16	SliceDeltaQPMin[2]				
		Format:	S7			
		Bange: [20 20]				
		Range: [-30,30]				
		I his field is the Slice level delta QP for bit-count below FrameBitRateMin - below 1/8				



MFX_MPEG2_PIC_STATE									
		and above 1/ 4This field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS control register when total bit count for the entire frame is between one-eighth and quarter the distance of FrameBitRateMinDelta from FrameBitRateMin, i.e., in the range of [(FrameBitRateMin- FrameBitRateMinDelta> (FrameBitRateMin- FrameBitRateMin- FrameBitRateMinDelta> 3)).					s a>>2),		
	15:8	SliceDeltaQPMin[1]							
		Forma	at:		S7				
		Range	[-30,30]						
		This field is the Slice level delta QP for bit-count below FrameBitRateMin- below 1/4 and above 1/ 2This field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS control register when total bit count for the entire frame is between quarter and half the distance of FrameBitRateMinDelta from FrameBitRateMin, i.e., in the range of [(FrameBitRateMin- FrameBitRateMinDelta>>1), (FrameBitRateMin- FrameBitRateMinDelta>>2)).							
	7:0	SliceDe	ltaQPMi	n[0]					
		Forma	t:			S7			
		Range: [-30,30]							
		This fic	Range. [-50,50]						
		2This field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS control register when total bit count for the entire frame is below FrameBitRateMin by more than half the distance of FrameBitRateMinDelta , i.e., in the range of [0, (FrameBitRateMin- FrameBitRateMinDelta >>1).							
10	31	FrameBitrateMaxUnit							
	This Val Oh 1h	This field is the Frame Bitrate Maximum Limit Units.							
[ExistsIf]Encode		Value	Name		Description				
Uniy		0h	Byte	FrameBitRateMax is in units of 32 Bytes when FrameBitrateMaxUnitMode is 1 and in units of 128 Bytes if FrameBitrateMaxUnitMode is 0					
		1h	Kilobyte	FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0					
	30	FrameBitrateMaxUnitMode							
	Bit V 0	BitFiel 1	tFiel This field is the Frame Bitrate Maximum Limit Units.dDesc						
		Value	Na	ame	Description				
		0h	Compatibility F mode		FrameBitRateMaxUnit is in old mo	ode (128b/16Kb)			
		1h	New mod	de	FrameBitRateMaxUnit is in new m (32byte/4Kb)	node			
	29:16	6 FrameBitRateMax This field is the Frame Bitrate Maximum Limit. This field along with FrameBitrateMaxI determines maximum allowed bits in a frame before multi-pass gets triggered (wher					MaxUnit vhen		


			MF	(_MPE	G2_PIC_STATE				
		enabled). In other words, multi-pass is triggered when the actual frame b exceeds this value. When FrameBitrateMaxUnitMode is 0(compatibility m should be used, bits 28 and 29 should be 0.					int ts 16:27		
		Valu	e Nam	e	Description				
		0-512k	(B	The proc	grammable range 0-512KB when	FrameBitrateMaxUr	nit is 0.		
		0-8190	KB	The prog	grammable range 0-8190KB whe	n FrameBitrateMaxL	Jnit is 1.		
	15	FrameE This fiel	BitrateMi d is the F	nUnit rame Bitrate	e Minimum Limit Units.				
		Value	Name		Description				
		0h Byte FrameBitRateMax is in units of 32 Bytes when FrameBitrateMinUnitMode is 1 and in units of 128 FrameBitrateMinUnitMode is 0			en of 128 Bytes if				
		1hKiloByteFrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes in FrameBitrateMaxUnitMode is 0			es when s of 16KBytes if				
	14	FrameBitrateMinUnitMode							
		Value	N	ame	Description				
		0h	compati mode	bility	FrameBitRateMaxUnit is in old r	meBitRateMaxUnit is in old mode (128b/16Kb)			
		1h	New Mo	de	FrameBitRateMaxUnit is in new (32byte/4Kb)	mode			
	13:0	FrameB This fiel determinent enablect than this should When F FrameB	ameBitRateMin is field is the Frame Bitrate Minimum Limit ()This field along with FrameBitrateMinUnit termines minimum allowed bits in a Frame before Multi-Pass gets triggered (when abled). In other words, multi-pass is triggered when the actual frame byte count is less an this value. When FrameBitrateMinUnitMode is 0 (compatibility mode) bits 0:11 ould be used, bits 12 and 13 should be 0. Range: The programmable range 0-512KB hen FrameBitrateMinUnit is in 0. Programmable range is 0-8190 KB when						
11	31	Reserve	ed						
[Evists]f]Encode		Format	:		MBZ				
Only	30:16	Frame	BitRateM	axDelta		1			
		Defaul	t Value:			0h			
		Access:				None			
		Format: U15							
		This fie shares The pr Bytes o This fie	eld is used the same ogramm or 16KB re eld is used	d to select th FrameBitra able range i espectively. d to select th	he slice delta QP when FrameBitF teMaxUnit. s either 0- 512KB or 4MBB in Fra he slice delta QP when FrameBitF	RateMax Is exceeded meBitrateMaxUnit c RateMax Is exceeded	d. It of 128 d. It		



		N	1FX_	MPEG2_PIC_STAT	E					
		shares the s 0(compatib	shares the same FrameBitrateMaxUnit. When FrameBitrateMaxUnitMode is 0(compatibility mode) bits 16:27 should be used, bits 28, 29 and 30 should be 0.							
	15	Reserved								
		Format:			MBZ					
	14:0	FrameBitRa This field is shares the s 0(compatibi requires the true, otherw Value 0-1024KB 0- 16380KB	iteMinE used to ame Fra lity moc followin rise it ma Name	Delta o select the slice delta QP when Fra meBitrateMinUnit. When FrameBit de) bits 0:11 should be used, bits 1 ng condition FrameBitRateMinDelt ay cause unpredicted behavior. Descr The programmable range 0-1024 in 32Bytes. Programmable range is 0-16380K 4Kbytes.	ameBitRateMin Is exceeded. It trateMinUnitMode is 2, 13 and 14 should be 0.Note: HW ta <= 2*FrameBitRateMinMust be ription KB When FrameBitrateMinUnit is KB when FrameBitrateMinUnit is in					
12	31:21	Reserved	1							
		Format:			MBZ					
	20	Reserved								
		Format:			MBZ					
	19	Reserved								
		Format: MBZ								
	18:16	Reserved	leserved							
		Format:			MBZ					
	15:0	Reserved								
		Format:			MBZ					



MFX_PAK_INSERT_OBJECT

Source:	
---------	--

VideoCS

2

Length Bias:

Description

The MFX_PAK_INSERT_OBJECT command is the first primitive command for the AVC and MPEG2 Encoding Pipeline.

This command is issued to setup the control and parameters of inserting a chunk of compressed/encoded bits into the current bitstream output buffer starting at the specified bit location perform the actual insertion by transferring the command inline data to the output buffer max, 32 bits at a time.

It is a variable length command as the data to be inserted are presented as inline data of this command. It is a multiple of 32-bit (1 DW), as the data bus to the bitstream buffer is 32-bit wide.

Multiple insertion commands can be issued back to back in a series. It is host software's responsibility to make sure their corresponding data will properly stitch together to form a valid H.264 bitstream.

Internally, MFX hardware will keep track of the very last two bytes' (the very last byte can be a partial byte) values of the previous insertion. It is required that the next Insertion Object Command or the next PAK Object Command to perform the start code emulation sequence check and prevention 0x03 byte insertion with this end condition of the previous insertion. Hardware will keep track of an output bitstream buffer current byte position and the associated next bit insertion position index. Data to be inserted can be a valid H.264 NAL units or a partial NAL unit. Certain NAL unit has a minimum byte size requirement. As such the hardware will optionally (enabled by STATE Command) determines the number of CABAC ZERO WORD to be inserted to the end of the current NAL, based on the minimum byte size of a NAL and the actual bin count of the encoded Slice. Since prior to the CABAC_ZERO_WORD insertion, the RBSP or EBSP is already byte-aligned, so each CABAC_ZERO_WORD insertion is actually a 3-byte sequence 0x00 00 03. The inline data may have already been processed for start code emulation byte insertion, except the possibility of the last 2 bytes plus the very last partial byte (if any). Hence, when hardware performing the concatenation of multiple consecutive insertion commands, or concatenation of an insertion command and a PAK object command, it must check and perform the necessary start code emulation byte insert at the junction. The inline data is required to be byte aligned on the left (first transmitted bit order) and may or may not be byte aligned on the right (last transmitted bits).

The command will specify the bit offset of the last valid DW. Each insertion state command defines a chunk of bits (compressed data) to be inserted at a specific location of the output compressed bitstream in the output buffer. Depend on CABAC or CAVLC encoding mode (from Slice State), PAK Object Command is always ended in byte aligned output bitstream except for CABAC header insertion which is bit aligned. In the aligned cases, PAK will perform 0 filling in CAVLC mode, and 1 filling in CABAC mode.

Insertion data can include:any encoded syntax elements bit data before the encoded Slice Data (PAK Object Command) of the current SliceSPS NALPPS NALSEI NALOther Non-Slice NALLeading_Zero_8_bits (as many bytes as there is)Start Code PrefixNAL Header ByteSlice



MFX_PAK_INSERT_OBJECT

HeaderAny encoded syntax elements bit data after the encoded Slice Data (PAK Object Command) of the current Slice and prior to the next encoded Slice Data of the next Slice or prior to the end of the bistream, whichever comes firstCabac_Zero_Word or Trailing_Zero_8bits (as many bytes as there is). Anything listed above before a Slice DataContext switch interrupt is not supported by this

Anything listed above before a Slice DataContext switch interrupt is not supported by this command.

DWord	Bit	Description									
0	31:29	Command Type	Command Type								
		Default Value:	3h PARALLEL_VIDEO_PIPE								
		Format:		ОрСо	de						
	28:27	Pipeline									
		Default Value:		2h MFX_	PAK_INSERT_OBJ	ECT					
		Format:		OpCode							
	26:24	Media Command Ope	code		-						
		Default Value:			0h MFX_COMMC	DN					
		Format:			OpCode						
	23:21	SubOpcode A									
		Default Value:				2h					
		Format:				OpCode					
	20:16	SubOpcode B									
		Default Value:				8h					
		Format:				OpCode					
	15:12	Reserved									
		Format:				MBZ					
	11:0	DWord Length	1								
		Default Value:	0h Exclud	es DWor	d (0,1) = Variable),1) = Variable Length in DW					
		Format:	=n Total I	_ength - 1	2						
1	31:18	Reserved									
		Format:				MBZ					
	17:16	DataByteOffset - Srcl Source Data Starting B	DataStarti vte Positio	ngByteO on within) ffset[1:0] the very first inlin	e DW.					
	15	HeaderLengthExclude	eFrmSize								
		In case this flag is on, bits are NOT accumulated during current access unit coding neither for									
		Cabac Zero Word Insel	rtion bits c	ounting (RAMF N(OF FOR OUTPUT IN IV	IMIO register					
		When using HeaderLe	enghtExclue	deFrmSiz	e for header inse	tion, the software needs to make sure					
		that data comes alread	ly with inse	erted star	rt code emulation	bytes. SW shouldn't set					
		EmulationFlag bit (Bit	3 of DWO	KD1 of N	IFX_PAK_INSERT_	UBJECT).					
		value	iame			Description					



	1.		FX_PAK_II	NSEK						
	1	NO_ACC	UMULATION	Bits duri	ng current call are not accumulated					
	0	ACCUMU	JLATE	All bits a	All bits accumulated					
14	Slice Header Indicator This bit indicates if the insert object is a slice header.									
	Value	Name			Description					
	1	SLICE_HEA	DER Insertion Obje HW and is use	ect is a Slic ed for inse	e Header. The command is stored internally by rting slice headers.					
	0	LEGACY	Legacy Inserti is not stored i	ion Object in HW.	command. The PAK Insertion Object command					
				Program	ning Notes					
	The pa NAL_ty The pa	yload for P/ pe and slice avload for P	AK_INS_OBJ should e header (slice_head AK INS OBJ should	contain o der() in AV dn't contai	nly start code for Slice header followed by 'C spec). n CABAC Byte alignment bits. HW adds these					
	alignment bits which are part of slice_data. Example PAK_INS_OBJ payload: 00 00 01 <nal_type> <slice_header_byte0> <slice_header_byte last=""> Any zero_bytes that are added before slice header can be inserted by any preceding general PAK INS_OBJ.</slice_header_byte></slice_header_byte0></nal_type>									
13:8	DataBitsInLastDW - SrCDataEndingBitInclusion[5:0] Source Data to be included in the very last inline DW. Follows the MSBit is the upper bit of each byte within the DW. The lower byte is actually processed first.For example, SrCDataEndingBitInclusion = 9, bit 7:0 and bit 15 are included as valid header data.									
	[1] 201		Value		Name					
	[1,32]									
7:4	SkipEm Skip em exampl	ulByteCnt nulation che e, to skip th	 Skip Emulation E ock for number of st e start code that ha 	Byte Coun tarting byt as already	i t esIt can be programmed from 0 to 15 bytes. Fo prefixed in the bitstream.					
3	Emulat	ionFlag - E	mulationByteBitsI	insertEnak	le					
	Value	Name			Description					
	0	NONE	No emulation							
	1	EMULATE	Instruct the hardware to perform Start Code Prefix (0x 00 00 01/02/03/00) Search and Prevention Byte (0x 03) insertion on the insertion data of this command. It is required that hardware will handle a start code prefix crossing the boundary between insertion commands, or an insertion command followed by a PAK Object command.							
2	LastHeaderFlag - LastSrcHeaderDataInsertCommandFlag To process a series of consecutive insertion commands, this flag (=1) indicates the current command is the last 'header' insertion in the series. In CABAC, hardware must perform the "1" insert for byte align for Slice Header before Slice Data comes in in the next PAK-OBJECT command In CAVIC bardware ignores this bit									
1	EndOfS No mor	liceFlag - L re insertion	.astDstDataInsertC	Command more PAK-	Flag OBJECT command follows. Flush data out to					

г



	MFX_PAK_INSERT_OBJECT											
		memory										
	0	BitstreamStartReset - ResetBitStreamStartingPos										
		OPEN:	This bit is	s redundant, the control is already in the Slice State command								
		Value Name Description										
		1	RESET	Reset the bitstream buffer insertion position to the bitstream buffer starting position.								
		0	INSERT	Insert the current command inline data starting at the current bitstream buffer insertion position								
2n	31:0	Insert Data PayLoad Actual Data to be inserted to the output bitstream buffer.										



MFX_PIPE_BUF_ADDR_STATE

Source: VideoCS

Length Bias:

This state command provides the memory base addresses for all row stores, StreamOut buffer and reconstructed picture output buffers required by the MFD or MFC Engine (that are in addition to the row stores of the Bit Stream Decoding/Encoding Unit (BSD/BSE) and the reference picture buffers). This is a picture level state command and is common among all codec standards and for both encoder and decoder operating modes. However, some fields may only applicable to a specific codec standard. All Pixel Surfaces (original, reference frame and reconstructed frame) in the Encoder are programmed with the same surface state (NV12 and TileY format), except each has its own frame buffer base address. In the tile format, there is no need to provide buffer offset for each slice; since from each MB address, the hardware can calculated the corresponding memory location within the frame buffer directly.

DWord	Bit		Description									
0	31:29	Command	Гуре	e								
		Default Val	ue:		3h PARALLEL_VIDEO_PIPE							
		Format:			OpCode							
	28:27	Pipeline										
		Default Val	ue:	2h	MFX_PIPE_BUF_ADDR_S	TATE						
		Format:		Op	Code							
	26:24	Common O	pcode									
		Default Val	ue:	0h	MFX_PIPE_BUF_ADDR_S	TATE						
		Format:		Op	Code							
	23:21	SubOpcode	e A									
		Default Val	ue:	0h	MFX_PIPE_BUF_ADDR_S	TATE						
		Format:										
	20:16	SubOpcode	B									
		Default Val	lue: 2h MFX_PIPE_BUF_ADDR_STATE									
		Format:		Op	Code							
	15:12	Reserved										
		Format:				MBZ						
	11:0	DWord Len	gth				T					
		Format:					=n					
		Total Longt	h									
		Fixed Lengt	:h									
		Tixed Leng										
		Value	Name Description									
		16h	DWORD_COUNT_n [Default] Excludes DWord (0,1)									
1	31:6	Pre Debloc	king - Destination	Ade	dress							
		Format:	Gra	phic	csAddress[31:6]							
		Specifies the	e 4K byte aligned fr	ame	e buffer address for outp	outting	the non-filtered reconstructed					



			MFX_	PIP	E_BUF_ADDR_STATE					
		YUV picture unit). This f	e (i.e. outpu ïeld is ignor	t of fina ed if Pr	al adder in each codec standard, and prior to eDeblockOutEnable is set to 0 (disable).	o the deblocking filter				
	5:4	Pre Deblocking - Arbitration Priority Control								
		Va	lue	ononty	Name	le for this surface.				
		00b		Highes	st priority					
		01b		Second	d highest priority					
		10b		Third h	nighest priority					
		11b		Lowest	priority					
	3:0	Reserved								
		Format:			MBZ					
2	31:6	Post Deblo	ocking - De	stinatio	on Address					
		Format:		Gr	aphicsAddress[31:6]					
		Specifies th reconstruct PostDebloc	ne 4K byte a ted YUV pic ckOutEnable	ligned f ture (i.e e is set t	frame buffer address for outputting the pos . output of the deblocking filter unit)This fie :o 0 (disable).	t-loop filtered eld is ignored if				
	5:4	Post Deblo This field co	ocking - Ar	bitratio priority	on Priority Control of arbitration used in the GAC/GAM pipelin	e for this surface.				
		Va	lue	Name						
		00b		Highes	Highest priority					
		01b		Second	Second highest priority					
		10b		Third h	hird highest priority					
		11b		Lowest	: priority					
	3	Reserved								
	2	Post Deblo This field co GTT. The eff is ignored f	ocking - Gra ontains the ffective GFD for reads.	a phics I GFDT b T is the	Data Type (GFDT) it for this surface when writes occur. GFDT of logical OR of this field with the GFDT from	can also be set by the the GTT entry. This field				
	1:0	Post Deblocking - Cacheability Control								
		Format:		J2 Enumerated type						
		This field c	ontrols cach	eability						
		Value	Nam	е	Description					
		006	GILentry		Use cacheability control bits from GTT enti	ry				
		116	Not cached	1	Data is not cached					
2	21.5		Cacheo							
3	31:6		ncompress	ed Pict	ure - Source Address (CurSrcAddr)					
		Exists IT:			Encounty					
		ronnat.		G	raphicsAudress[51.0]					



			MFX	_PIP	E_BUF_ADDR_STATE				
		Specifies the 64 byte aligned frame buffer address for fetching YUV pixel data from the original uncompressed input picture for encoding.							
	5:4	Original U	ginal Uncompressed Picture - Arbitration Priority Control						
		This field co	ontrols the	priority of arbitration used in the GAC/GAM pipeline for this surface.					
		Va	lue		Name				
		00b		Highes	t priority				
		01b		Second	highest priority				
		10b		Third h	ighest priority				
-		11b		Lowest	priority				
	3	Reserved							
	2	Original U	ncompress	ed Pictu	ure - Graphics Data Type (GFDT)				
		This field co	ontains the	GFDI bi Tis the	It for this surface when writes occur. GFD1 can also be set by the				
		is ignored f	for reads.		logical of of this licit with the of br from the off entry. This licit				
	1:0	Original U	ncompress	ed Pictu	ure - Cacheability Control				
		This field co	ontrols cach	neability					
		Value	Nam	e	Description				
		00b	GTT entry		use cacheability control bits from GTT entry				
		01b	Not cache	d	data is not cached				
		11b	Cached		Data is cached				
4	31:6	StreamOut	t Data Dest	tination	- Base Address (StreamOutAddr)				
		Format:		Gr	aphicsAddress[31:6]				
		Specifies th StreamOut	ne 64 byte a Enable is se	ligned a t in the	address for outputting the per-MB indirect data to memory when MFX_PIPE_MODE_SELECT command.				
		For encode	er: this field	is used t	for dynamic repeat of frame in PAK for Rate Control. Also used for				
		feeding co	ding inform	ation ba	ack to the Host, Video Preprocessing Unit and ENC Unit.All data are				
		written in f	ixed format	s, and th	nerefore all record sizes are known in the hardware. Hardware can				
			ie onset int	o this da	ase address for per-MB data.				
	5:4	StreamOut	t Data Dest	ination	- Arbitration Priority Control				
		This field co	ontrols the	priority	of arbitration used in the GAC/GAM pipeline for this surface.				
		Va	lue		Name				
		00b		Highes	t priority				
	01b Second highest priority								
		10b		Third h	ighest priority				
		11b		Lowest	priority				
	3	Reserved							
	2	StreamOut This field co	t Data Dest ontains the	GFDT bi	- Graphics Data Type (GFDT) it for this surface when writes occur. GFDT can also be set by the				



			MFX_F	PIP	E_BUF_ADDR_STATE						
		GTT. The ef	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	StreamOu This field co	t Data Destina ontrols cachea	ation ability	- Cacheability Control						
		Value	Name	<u></u>	Description						
		00b	GTT entry		use cacheability control bits from GTT entry						
		01b	Not cached		data is not cached						
		11b	Cached		Data is cached						
5	31:6	Intra Row	Store Scratch	n Buff	er - Base Address (IntraOSRowStoreAddr)						
		Format:		Gra	aphicsAddress[31:6]						
		This field p IntraPredic macroblocl Hardware u This field is either MBA	This field provides the base address of the scratch buffer (read/write) used by the AVC IntraPrediction unit to store MB information of the previous row for processing of each macroblock in the current row. The Intra Row Store buffer must be 64-byte cacheline aligned. Hardware uses the horizontal address of the current macroblock to address the Intra Row Store. This field is ignored in MPEG2 and VC1 mode. Max 256 cachelines for 4K pixels (1 cacheline for either MBAFF or non-MBAFF)								
	5:4	Intra/Over	lap Smoothir ontrols the pri	n g Ro ority (w Store Scratch Buffer - Arbitration Priority Control of arbitration used in the GAC/GAM pipeline for this surface.						
		Va	lue		Name						
		00b	Н	ighes	t priority						
		01b	Se	econd	highest priority						
		10b	Tł	hird h	ighest priority						
		11b	Lo	owest	priority						
	3	Reserved									
	2	Intra/Over This field co GTT. The eff is ignored f	·lap Smoothir ontains the GF ffective GFDT i for reads.	ng Ro DT bi is the	w Store Scratch Buffer - Graphics Data Type (GFDT) t for this surface when writes occur. GFDT can also be set by the logical OR of this field with the GFDT from the GTT entry. This field						
	1:0	Intra/Over This field co	lap Smoothir ontrols cachea	n g Ro ability.	w Store Scratch Buffer - Cacheability Control						
		Value	Name		Description						
		00b	GTT entry		use cacheability control bits from GTT entry						
		01b	Not cached		data is not cached						
		11b	Cached		Data is cached						
6	31:6	Deblockin	g Filter Row S	Store	Scratch Buffer - Base Address (DeblockRowStoreAddr						
		Format:		Gra	aphicsAddress[31:6]						
		Deblocking	Filter Row Sto	ore is	needed for						
		VC1 Overla	LIN-LOOP De	Filter	ang ritter						
		This field p	rovides the ba	ase ad	dress of the scratch buffer (read and write) used by the deblocking						
		filter unit to store MB information of the previous row for filtering of each macroblock in the									



			MFX	PIP	E_BUF_ADDR_STATE			
		current row. The Deblocking Filter Row Store buffer must be 64-byte cacheline aligned. Hardware uses the horizontal address of the current macroblock to address the Deblocking Row Store. Max 6 cachelines for VC1 and MPEG2, and max 4 for AVC (for MBAFF, 2 for non MBAFF).						
	5:4	Deblocking Filter Row Store Scratch Buffer - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface						
		Va	lue	<u>-</u>	Name			
		0h		Highes	t priority			
		1h		Second	highest priority			
		2h		Third h	ighest priority			
		3h		Lowest	priority			
	3	Reserved						
	2	Deblockin This field co GTT. The et is ignored	g Filter Rov ontains the ffective GFD for reads.	v Store GFDT bi T is the	Scratch Buffer - Graphics Data Type (GFDT) it for this surface when writes occur. GFDT can also be set by the logical OR of this field with the GFDT from the GTT entry. This field			
	1:0	Deblockin This field c	g Filter Rov	v Store eability	Scratch Buffer - Cacheability Control			
		Value	Nam	е	Description			
		00b	GTT entry		use cacheability control bits from GTT entry			
		01b	Not cached	ł	data is not cached			
		11b	Cached		Data is cached			
722	31:6	Reference	Picture (Re	fAddr[0-15]) - Addresses			
		Format:	ha C4 huta i	Gr	aphicsAddress[31:6]			
	Specifies the 64 byte aligned reference frame buffer addresses for the motion compensate operation in AVC/VC1/MPEG2. AVC can specify up to 16 YUV frame-based surfaces for both forward and backward references, i.e. L0+L1 total = 16 max. Any entry can be assigned to L1 or both lists. But VC1 and MPEG2, worst case, can use up to 2 YUV frame-based surfaces both forward and backward references:P-MB: RefAddr[0] - temporal closest previous field reference frame (can be the current frame)RefAddr[1] - next temporal closest previous field reference frame (must be different from the current frame)It is a variant (without the LongTermRefPic specification) of the RefFrameList[16] defined in AVC DXVA Spec. RefAdd is indexed by frame_storeID >>1. It is not a packed list, i.e. invalid entries can scatter amon list. All invalid addresses must be set to a valid address RefAddr[0] by the driver. The same applies to VC1 and MPEG2.							
					Programming Notes			
		AVC: Alwa max num condition,	ys specifies of active ref etc.) by hav	all 16 a erence ing all t	ddresses even some of them are not needed as indicated by the pictures. This is done for preventing data corruption (error, fault the references being set to a legal location.			
	5:4	Reference This field c	Picture (Re ontrols the p	fAddr[priority	0-15]) - Arbitration Priority Control of arbitration used in the GAC/GAM pipeline for this surface.			



MFX PIPE BUF ADDR STATE Value Name 00b Highest priority Second highest priority 01b 10b Third highest priority 11b Lowest priority Reserved 3 2 Reference Picture (RefAddr[0-15]) - Graphics Data Type (GFDT) 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. H/W only reads this bit from the very first RefAddr[0][bit 3:0], all other RefAddr[i][bit 3:0] are ignored by H/W and are assumed to have the same values as that of RefAddr[0]. 1:0 Reference Picture (RefAddr[0-15]) - Cacheability Control This field controls cacheability. H/W only reads this bit from the very first RefAddr[0][bit 3:0], all other RefAddr[i][bit 3:0] are ignored by H/W and are assumed to have the same values as that of RefAddr[0]. Value Name Description use cacheability control bits from GTT entry 00b GTT entry 01b Not cached data is not cached Data is cached 11b Cached 23 31:6 Macroblock Status Buffer Base Address (MacroblockStatAddr) Format: GraphicsAddress[31:6] Specifies the 64 byte aligned address for reading the per-MB indirect data from memory when MacroblockStatEnable is set in the MFX_AVC_IMG_STATE Command. For decoder: this field is ignored by hardware. For encoder: this field is used for dynamic repeat of frame in PAK for Rate Control. Also used for feeding coding information back to the Host, Video Preprocessing Unit and ENC Unit.All data are written in fixed formats, and therefore all record sizes are known in the hardware. Hardware can calculate the offset into this base address for per-MB data. **Arbitration Priority Control** 5:4 This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. Value Name 00b Highest priority 01b Second highest priority 10b Third highest priority 11b Lowest priority 3 Reserved 2 **Graphics Data Type (GFDT** 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.



	MFX_PIPE_BUF_ADDR_STATE									
	1:0	Cacheabili This field c	Cacheability Control This field controls cacheability.							
		Value	Value Name Description							
		00b	GTT	use cacheability control bits from GTT entry						
		01b	Not cached	data is not cached						
		11b	Cached	Data is cached						
24	31:0	Reserved								
		Format:			MBZ					



MFX	PIPE	MODE	SELECT
_	_		

Source: VideoCS

Length Bias:

Specifies which codec and hardware module is being used to encode/decode the video data, on a perframe basis.

The MFX_PIPE_MODE_SELECT command specifies which codec and hardware module is being used to encode/decode the video data, on a per-frame basis. It also configures the hardware pipeline according to the active encoder/decoder operating mode for encoding/decoding the current picture. Commands issued specifically for AVC and MPEG2 are ignored when VC1 is the active codec.

DWord	Bit		Description						
0	31:29	Command	Туре						
		Default Val	ue:		3h PA	3h PARALLEL_VIDEO_PIPE			
		Format:			ОрСо	de			
	28:27	227 Pipeline							
		Default Val	ue:			2h MFX_COMMON			
		Format:	nat:			OpCode			
	26:24	Opcode							
		Default Val	ue:		0h MF	X_COMMON_STA	TE		
		Format:			OpCod	de			
	23:21	SubOpA							
		Default Val	ue:				0h		
		Format:				OpCode			
	20:16	SubOpB						1	
		Default Val	ue:	(0h MFX	_PIPE_MODE_SELI	CT		
		Format:		(OpCode	pCode			
	15:12	Reserved							
		Format:					MBZ		
	11:0	DWord Len	igth						
		Format:		=n T	Total Le	ngth - 2			
		Value		N	Jamo		Description		
		3h	DWORD COUNT	n ID	Default1		Excludes DWord (0.1)		
1	21	Reserved							
-	30	Reserved							
	29	Reserved							
	28:27	Reserved							
	26:25	Reserved							
		Format:					MBZ		
	24	Reserved							



22.13	Reserved								
	Format: MBZ								
17	Decoder	Short Format N	lode						
	For IT mo	ode, this bit must	be 0.						
	Value	N	lame	Description					
	1 I	Long Format Drive	er Interface	AVC/VC1/MVC Long Format Mode is in use.					
	0 9	Short Format Driv [Default]	er Interface	AVC/VC1/MVC Short Format Mode is in use					
16:15	Decoder Each cod field sele	Mode select ling standard sup ects which one is i	ports two entry poin n use. This field is or	its: VLD entry point and IT (IDCT) entry point. The second s					
	Value	Name		Description					
	0h	VLD Mode	All codec minimum must support this mode Configure the MFD Engine for VLD Mode Note: All codec minimum must support this mode						
	1h	IT Mode	Configure the MFD Note: Only VC1 and	D Engine for IT Mode nd MPEG2 support this mode					
14:13	Reserved Format: MBZ								
12	Reserve	d							
	Format:			MBZ					
	THE PROPERTY AND	y status Report L	indule.						
**	This field decoder DW3 alo "Decode Commar otherwis In encod	I control whether modes: Error repond ng with the VLD/ d Picture Error/St d. Note: driver sh e, hardware migh er modes: Not us	the error/status repo orting is written out o IT error status bits ar atus Buffer address" nall program differen t overwrite previous ed	orting is enable or not.0: Disable1: EnableIn once per frame. The Error Report frame ID liste re packed into one cache and written to the listed in the MFX_PIPE_BUF_ADDR_STATE at error buffer addresses between pictrues; written data if driver does not read it fast enou					
**	This field decoder DW3 alo "Decode Commar otherwise In encod	I control whether modes: Error repo ng with the VLD/I d Picture Error/St nd. Note: driver sh e, hardware migh er modes: Not us Value	the error/status repo orting is written out o IT error status bits ar atus Buffer address" nall program differen t overwrite previous ed	orting is enable or not.0: Disable1: EnableIn once per frame. The Error Report frame ID liste re packed into one cache and written to the listed in the MFX_PIPE_BUF_ADDR_STATE at error buffer addresses between pictrues; written data if driver does not read it fast enou Name					
	This field decoder DW3 alo "Decode Commar otherwise In encod	l control whether modes: Error repo ng with the VLD/I d Picture Error/St nd. Note: driver sh e, hardware migh er modes: Not us Value	the error/status repo orting is written out o IT error status bits ar atus Buffer address" nall program differen t overwrite previous ed	orting is enable or not.0: Disable1: EnableIn once per frame. The Error Report frame ID liste re packed into one cache and written to the listed in the MFX_PIPE_BUF_ADDR_STATE at error buffer addresses between pictrues; written data if driver does not read it fast enou Name Disable					
	This field decoder DW3 alo "Decode Commar otherwise In encod 0h 1h	l control whether modes: Error repo ng with the VLD/I d Picture Error/St nd. Note: driver sh e, hardware migh er modes: Not us Value	the error/status repo orting is written out o IT error status bits ar atus Buffer address" nall program differen t overwrite previous ed	orting is enable or not.0: Disable1: EnableIn once per frame. The Error Report frame ID liste re packed into one cache and written to the listed in the MFX_PIPE_BUF_ADDR_STATE at error buffer addresses between pictrues; written data if driver does not read it fast enou Name Disable					
10	This field decoder DW3 alo "Decode Commar otherwise In encod Oh 1h Stream - This field decoding	I control whether modes: Error repo ng with the VLD/I d Picture Error/St nd. Note: driver sh e, hardware migh er modes: Not us Value Out Enable I controls whether g for transcoding	the error/status repo orting is written out of IT error status bits ar atus Buffer address" hall program differen t overwrite previous ed	orting is enable or not.0: Disable1: EnableIn once per frame. The Error Report frame ID liste re packed into one cache and written to the listed in the MFX_PIPE_BUF_ADDR_STATE at error buffer addresses between pictrues; written data if driver does not read it fast enou Name Disable Enable					
10	This field decoder DW3 alo "Decode Commar otherwis In encod 0h 1h Stream - This field decoding	I control whether modes: Error repo ng with the VLD/I d Picture Error/St nd. Note: driver sh e, hardware migh er modes: Not us Value Out Enable I controls whether g for transcoding Value	the error/status repo orting is written out of IT error status bits ar atus Buffer address" hall program differen t overwrite previous ed E r the macroblock par purpose.	orting is enable or not.0: Disable1: EnableIn once per frame. The Error Report frame ID liste re packed into one cache and written to the listed in the MFX_PIPE_BUF_ADDR_STATE at error buffer addresses between pictrues; written data if driver does not read it fast enou Name Disable Enable rameter stream-out is enabled during VLD Name					
10	This field decoder DW3 alo "Decode Commar otherwise In encod Oh 1h Stream - This field decoding	I control whether modes: Error repo ng with the VLD/I d Picture Error/St nd. Note: driver sh e, hardware migh er modes: Not us Value	the error/status repo orting is written out of IT error status bits ar atus Buffer address" hall program differen t overwrite previous ed r the macroblock par purpose.	orting is enable or not.0: Disable1: EnableIn once per frame. The Error Report frame ID liste re packed into one cache and written to the listed in the MFX_PIPE_BUF_ADDR_STATE at error buffer addresses between pictrues; written data if driver does not read it fast enou Name Disable Enable rameter stream-out is enabled during VLD Name Disable					
10	This field decoder DW3 alo "Decode Commar otherwise In encod Oh 1h Stream- This field decoding Oh 1h	I control whether modes: Error repo ng with the VLD/I d Picture Error/St nd. Note: driver sh e, hardware migh er modes: Not us Value	the error/status reporting is written out of orting is written out of IT error status bits ar atus Buffer address" hall program differen t overwrite previous ed c c c c c c c c c c c c c c c c c c	orting is enable or not.0: Disable1: EnableIn once per frame. The Error Report frame ID liste re packed into one cache and written to the listed in the MFX_PIPE_BUF_ADDR_STATE at error buffer addresses between pictrues; written data if driver does not read it fast enou Name Disable Enable rameter stream-out is enabled during VLD Name Disable Enable					
10	This field decoder DW3 alo "Decode Commar otherwise In encod Oh 1h Stream - This field decoding Oh 1h	I control whether modes: Error repo ng with the VLD/I d Picture Error/St nd. Note: driver sh e, hardware migh er modes: Not us Value	the error/status reporting is written out of IT error status bits ar atus Buffer address" hall program different t overwrite previous ed t the macroblock par purpose.	orting is enable or not.0: Disable1: EnableIn once per frame. The Error Report frame ID liste re packed into one cache and written to the listed in the MFX_PIPE_BUF_ADDR_STATE at error buffer addresses between pictrues; written data if driver does not read it fast enou Name Disable Enable rameter stream-out is enabled during VLD Name Disable Enable					



				MFX_	PI	PE_	MO	DE_SEL	ECT
		the inp re-con confor use thi PAK. T	the input compressed stream, selected decoded information may be used by the encoder for re-compression. In encoder modes: This feature used to perform dynamic Multipass of PAK for conformance pupose. Also it provides feedback to host (ENC) for future needs. Software can use this bit to disable writing PAK steam data to the streamout buffer for last pass of frame in PAK. Thus, save memory bandwidth.						
	9	Post Deblocking Output Enable (PostDeblockOutEnable) This field controls the output write for the reconstructed pixels AFTER the deblocking filter. In MPEG2 decoding mode, if this is enabled, VC1 deblocking filter is used.							e) ixels AFTER the deblocking filter. In filter is used.
				Value				Ŭ	Name
		0h						Disable	
		1h						Enable	
	8	Pre De This fie	block ld coi	king Output E ntrols the outp	nabl out w	l e (Pre l vrite for	Debloc the re	kOutEnable) constructed p	ixels BEFORE the deblocking filter.
				Value					Name
		Oh						Disable	
		1h						Enable	
	7:6	Reserv	ed						
		Forma	t:						MBZ
	5	Stitch	Mode	9					
		Exists I	lf:	//CodecS	Sel=E	Incode	and St	andardSel=A	VC
						Γ			
		Value	NL-4	Name		Description			
		UN 1 h	INOT	In stitch mode	2	This was done when we do for any Condersor by some hitford			
		T U	moc	le special stitci	1	conditions are met.			
	4	Codec	Selec	t					
		Valu	ue	Name		Description			
		0h		Decode					
		1h		Encode	Va	lid only	/ if Star	idardSel is AV	C, MPEG2)
	3:0	Standa	rd Se	elect					
		Va	alue	Na	me				Description
		0000b		MPEG2					
		0001b		VC1					
		0010b		AVC			Cover	s both AVC ar	nd MVC
		0011b		JPEG					
		0110b		Reserved					
0111b Reserved									
2	31	Reserv	ed						
		Forma	t:						MBZ



MFX	PIPE	MODE	SELECT

30	Reserved							
50	Format:		MBZ					
29	Reserved							
23	Format:		MBZ					
28.26	5 Reserved							
20.20	Format:		MBZ					
25	Reserved							
25	Format:		MBZ					
24	Reserved							
	Format:		MBZ					
23	Reserved							
23	Format:		MBZ					
22.21	Reserved							
20	Reserved							
19	Reserved							
18	Reserved							
	Format:		MBZ					
17	Reserved		I					
	Format:		MBZ					
16	Reserved		I					
15	Reserved							
14	Reserved							
	Format:		MBZ					
13	Reserved							
	Format:		MBZ					
12	Reserved							
	Format:		MBZ					
11	Reserved							
10	MPC pref08x8_disable Flag (Def	ault 0)						
	Value		Name					
	0 Disable							
	1	Enable						
9	Reserved							
	Format:		MBZ					
8	Reserved							
	Format:	MBZ						
7	Reserved							



				MFX_	PIPE_MODE_S	EL	ECT				
	6	Clock ga BitField	ate Enab Desc:	le at Slice	-level						
		Value Name Description									
		0 Disable Disable Slice-level Clock gating, Unit-level Clock gating will apply									
		1	Enable	Enable S	lice-level Clock gating, ov	erride	es any Unit level Clock gating				
	5	Reserve	ed								
	4	Reserve	d								
	3	Reserve	d								
	2	Reserve	d								
	1:0	Reserve	d								
		Format					MBZ				
3	31:0	Pic Stat	us/Error	Report ID							
		Exists If	:		//Decoder Mode Only						
		Format			U32						
		In decod	ler mode	s: Error rep	porting is written out once	per	frame. This field along with the VLD				
		"Decode	ed Picture	e Error/Stat	us Buffer address" listed i	n to n the	MFX PIPE BUE ADDR STATE				
		Comma	nd.								
		Val	ue		Name		Description				
		0h		32-bit unsi	gned	Unic	que ID Number				
		1h	1	Reserved	eserved						
4	31:0	Reserve	d								
		Format					MBZ				



VideoCS

2

MFX_QM_STATE

Source:

Length Bias:

This is a common state command for AVC encoder modes. For encoder, it represents both the forward QM matrices as well as the decoding QM matrices. This is a Frame-level state. Only Scaling Lists specified by an application are being sent to the hardware. The driver is responsible for determining the final set of scaling lists to be used for decoding the current slice, based on the AVC Spec Table 7-2 (Fall-Back Rules A and B). In MFX AVC PAK mode, PAK needs both forward Q scaling lists and IQ scaling lists. The IQ scaling lists are sent as in MFD in raster scan order. But the Forward Q scaling lists are sent in column-wise raster order (column-by-column) to simplify the H/W. Driver will perform all the scan order conversion for both ForwardQ and IQ.

DWord	Bit		Description					
0	31:29	Command T	уре					
		Default Valu	e:		3h P	ARALLEL_VIDEO_	VIDEO_PIPE	
		Format:			OpC	ode		
	28:27	Pipeline						
		Default Valu	efault Value: 2h MFX_MUL				DW	
		Format:				OpCode		
	26:24	Media Command Opcode						
		Default Valu	e:		0h M	FX_COMMON_S1	ΓΑΤΕ	
		Format:			OpCo	ode		
	23:21	SubOpcode	Α					
		Default Valu	e:				0h	
		Format:					OpCode	
	20:16	SubOpcode	В					
		Default Value:					7h	
		Format:					OpCode	
	15:12	Reserved				1		
		Format:					MBZ	
	11:0	DWord Leng	jth		1			
		Default Valu	e:		20h I	Excludes DWord	(0,1)	
		Format:			=n T	otal Length - 2		
1	31:2	Reserved						
		Format:					MBZ	
	1:0	MPEG2		1				
		Exists If:						
		For MPEG2 (zer Matrix is loaded.					
		Value				Name	3	
		0	MPEG_IN	FRA_QUA	NTIZE	R_MATRIX		
		1	MPEG_NC	N_INTRA	A_QUA	NTIZER_MATRIX		
		2-3	Reserved					



MFX_QM_STATE										
	1:0	AVC	AVC							
		Exists If		//AVC- Decoder Only						
		For AVC	QM Type : This fie	ld specifies which Quantizer N	latrix is loaded.					
		Value		Name						
		0	0 AVC_4x4_Intra_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)							
		1	AVC_4x4_Inter_MA	ATRIX, (Y-4DWs, Cb-4DWs, Cr-	4DWs, reserved-4DWs)					
		2	AVC_8x8_Intra_MA	ATRIX						
		3	AVC_8x8_Inter_MA	ATRIX						
233	31:0	Forward	d Quantizer Matrix	1						
		Format:	Format: U32							
		The form unsigned	The format of a Quantizer Matrix is an 8x8 matrix in raster order. Each element is an unsigned byte.							



MFX STATE POINTER

VideoCS Source: 2

Length Bias:

The MFX_STATE_POINTER command, issued at picture level, is used to set up the indirect pointers for VCS to fetch all the MFX states (Image state, Slice state, etc.) needed for the encoding/decoding process in PAK/IT mode. The encoding/decoding states are presented by state commands, which are grouped into separate sets (picture level, slice level, etc.), and each is stored in its own memory buffer referred by an indirect state pointer. The content of each indirect state buffer is a list of MFX state commands with no special format requirements. The sequence of commands in each indirect state buffer is terminated by a MI_BATCH_BUFFER_END command (acts as the last command marker). Therefore, indirect state buffers can have different and variable length of command sequences. The indirection is designed to facilitate context switching in the middle of a codec operation. The smallest granularity of interruption is designed to be at a completed MB row in AVC/VC1/MPEG2 IT and AVC PAK operating modes as well as in VC1/MPEG2 VLD mode. There is no support for context switch in AVC VLD mode.

Hardware supports up to 4 separate indirect state pointers, allowing software to manage the grouping of state commands. During context switch, hardware restores (re-issues) the latest version of each indirect state pointer, if present.

MFX_STATE_POINTER command can only program one indirect state pointer at a time. MI_FLUSH will invalidate all indirect state buffer pointers inside VCS.

DWord	Bit	Description							
0	31:29	Command Type							
		Default Value:		3h G	FX_PIPE				
		Format:		ОрСо	ode				
	28:27	Pipeline							
		Default Value:			2h Media				
		Format:			OpCode				
	26:24	Media Command Opcode	Aedia Command Opcode						
		Default Value:	0h MFX_COMM	10N_9	TATE				
		Format:	OpCode						
	23:21	SubOpcode A							
		Default Value:		0h					
		Format:			OpCode				
	20:16	SubOpcode B							
		Default Value:			6h				
		Format:			OpCode				
	15:12	Reserved							
		Format:			MBZ				
	11:0	DWord Length							
		Default Value:	0h DWORD	RD_COUNT_n					



			MF	K_STATE_POINTER	R				
		Format: =n Total Length - 2							
1	31:5	State Pointer							
		Format:	ormat: GeneralStateOffset[31:5]Indirect State Buffer						
		Specifies th	e 32-byte a	ligned address of an Indirect State	e Buffer. This pointer is	relative to			
		the Genera	State Base	Address.					
	4:2	Reserved							
		Format:			MBZ				
	1:0	State Point	ter Index						
		Specifies or	ne of the fo	ur indirect state pointers to progra	am.	1			
		Value	Name	Description	n				
		00b		indirect state pointer 0 (image sta	ate)				
01b indirect state pointer 1 (slice state)sc									
		10b		indirect state pointer 2					
		11b		indirect state pointer 3					



MFX_STITCH_OBJECT

Source: VideoCS

2

Length Bias:

The MFC_STITCH_OBJECT command is used when stitch-enabled is set to 1, while CodecSel and StandardSel are set to ENCODE and AVC, respectively. This command is used, for example, to stitch multiple bitstreams to form a transport stream.

It is a variable length command as the data to be inserted are presented as either inline data and/or indirect data of this command. Multiple insertion commands can be issued back to back in a series. It is host software's responsibility to make sure their corresponding data will properly stitch together to form a valid output. Hardware keeps track of an output bitstream buffer current byte position and the associated next bit insertion position index.

DWord	Bit	Description						
0	31:29	Command Type						
		Default Value:	:	3h PA	RALLEL_VIDEO_P	LLEL_VIDEO_PIPE		
		Format: Op0			ode			
	28:27	Pipeline						
		Default Value:			AFC_STITCH_OBJE	CT		
2		Format:		ОрС	ode			
	26:24	Media Command	Opcode					
		Default Value:			0h MFX_COMM	ON		
		Format:			OpCode			
	23:21	SubOpcode A						
		Default Value:				2h		
-		Format:				OpCode		
	20:16	SubOpcode B	SubOpcode B					
		Default Value:				Ah		
		Format:				OpCode		
	15:12	Reserved						
		Format:				MBZ		
	11:0	DWord Length						
		Default Value:	0h Excludes DW	ord (0,1) = Variable Le	ngth in DW (>= 3)		
		Format:	=n Total Length	- 2				
		If it is 3, it indicates	s the absent of inl	ine da	ata.			
1	31:18	Reserved						
		Format:				MBZ		
	17:16	Source Data Start	ing Byte Offset					
		Source Data Starti	ng Byte Position w	<i>ithin</i>	the very first inlin	ie DW.		
-	15:14	Reserved						



		MFX_STITCH_C	OBJECT				
		Format:	MBZ				
	13:8	Source Data Ending Bit Inclusion Source Data to be included in the very last inline DW. Follows the MSBit is the upper bit of each byte within the DW. The lower byte is actually processed first.For example, SrCDataEndingBitInclusion =9, bit 7:0 and bit 15 are included as valid header data.					
		Value	Name				
		[1,32]					
	7:4	Reserved					
	3	Reserved					
	2	Last Source Header Data Insert Command Flag To process a series of consecutive insertion commands, this flag (=1) indicates the current command is the last 'header' insertion in the series. In CABAC, hardware must perform the "1" insert for byte align for Slice Header before Slice Data comes in in the next PAK-OBJECT command. In CAVLC, bardware ignores this bit					
	1	Last Destination Data Insert Command Flag					
	Header Data Insert Command Flag						
		No more insertion command and no more PAK-OBJECT command follows. Flush data out memory					
	0	Reserved					
2	31:19	Reserved					
		Format:	MBZ				
	18:0	Indirect Data Length					
		Format:	U19				
		This field provides the length in bytes of the ind data fetching is disabled - subsequently, the Inc field must have the same alignment as the Indir	lirect data. A value zero indicates that indirect lirect Data Start Address field is ignored. This ect Object Data Start Address.				
3	31:0	Indirect Data Start Address					
		Format: MfxIndirectBitstreamObjectAd	ldress[31:0]				
		This field specifies the Graphics Memory starting for processing. This pointer is relative to the MF Hardware ignores this field if indirect data is not	field specifies the Graphics Memory starting address of the data to be loaded into the kernel rocessing. This pointer is relative to the MFX Indirect Bitstream Object Base Address. ware ignores this field if indirect data is not present.				
4n	31:0	Insert Data PayLoad Inline data to be inserted to the output bitstream	m buffer				



MFX_SURFACE_STATE

Source: VideoCS

2

Length Bias:

This command is common for all encoding/decoding modes, to specify the uncompressed YUV picture (i.e. destination surface) or intermediate streamout in/out surface (e.g. coefficient/residual) (field, frame or interleaved frame) format for reading and writing:

- Uncompressed, original input picture to be encoded
- Reconstructed non-filtered/filtered display picturec(becoming reference pictures as well for subsequent temporal inter-prediction)

Since there is only one media surface state being active during the entire encoding/decoding process, all the uncompressed/reconstructed pictures are defined to have the same surface state. The primary difference among picture surface states is their individual programmed base addresses, which are provided by other state commands and not included in this command. MFX engine is making the association of surface states and corresponding buffer base addresses.

MFX engine currently supports only one media surface type for video and that is the NV12 (Planar YUV420 with interleaved U (Cb) and V (Cr). For optimizing memory efficiency based on access patterns, only TileY is supported. For JPEG decoder, only IMC1 and IMC3 are supported. Pitch can be wider than the Picture Width in pixels and garbage will be there at the end of each line. The following describes all the different formats that are supported and not supported in MFX :

- NV12 4:2:0 only; UV interleaved; Full Pitch, U and V offset is set to 0 (the only format supported for video codec); vertical UV offset is MB aligned; UV xoffsets = 0. JPEG does not support NV12 format because non-interleave JPEG has performance issue with partial write (in interleaved UV format)
- IMC 1 & 3 Full Pitch, U and V are separate plane; (JPEG only; U plane + garbage first in full pitch followed by V plane + garbage in full pitch). U and V vertical offsets are block aligned; U and V xoffset = 0; there is no gap between Y, U and V planes. IMC1 and IMC3 are different by a swap of U and V. This is the only format supported in JPEG for all video subsampling types (4:4:4, 4:2:2 and 4:2:0)
- We are not supporting IMC 2 & 4 Full Pitch, U and V are separate plane (JPEG only; U plane first in full pitch followed by V plane in full pitch U and V plane are side-by-side). U and V vertical offsets are 16-pixel aligned; V xoffset is half-pitch aligned; U xoffset is 0; there is no gap between Y, U and V planes. IMC2 and IMC4 are different by a swap of U and V.
- We are not supporting YV12 half pitch for each U and V plane, and separate planes for Y, U and V (U plane first in half pitch followed by V plane in half pitch). For YV12, U and V vertical offsets are block aligned; U and V xoffset = 0; there is no gap between Y, U and V planes

Note that the following data structures are not specified through the media surface state

- 1D buffers for row-store and other miscellaneous information.
- 2D buffers for per-MB data-structures (e.g. DMV biffer, MB info record, ILDB Control and Tcoeff/Stocoeff).

This surface state here is identical to the Surface State for deinterlace and sample_8x8 messages described in the Shared Function Volume and Sampler Chapter.

For non pixel data, such as row stores, indirect data (Compressed Slice Data, AVC MV record, Coeff record and AVC ILDB record) and streamin/out and output compressed bitstream, a linear buffer is employed. For row stores, the H/W is designed to guarantee legal memory accesses (read and write). For the remaining cases, indirect object base address, indirect object address upper bound, object data start address (offset) and object data length are used to fully specified their corresponding buffer. This



MFX_SURFACE_STATE

mechanism is chosen over the pixel surface type because of their variable record sizes. All row store surfaces are linear surface. Their addresses are programmed in Pipe_Buf_Base_State or Bsp_Buf_Base_Addr_State

Programming Notes

VC1 I picture scaling: Even though VC1 allows I reconstructed picture scaling (via RESPIC), as such scaling is only allowed at I picture. All subsequent P (and B) pictures must have the same picture dimensions with the preceding I picture. Therefore, all reference pictures for P or B picture can share the same surface state with the current P and B picture. Note: H/W is not processing RESPIC. Application is no longer expecting decoder pipeline and kernel to perform this function, it is going to be done in the video post-processing scaler or display controller scale as a separate step and controller.

All video codec surfaces must be NV12 Compliant, except JPEG. U/V vertical must be MB aligned for all video codec (further contrained for field picture), but JPEG can be block aligned. All video codec and JPEG uses Tiled - Y format only, for uncompressed pixel surfaces.

Even for JPEG planar 420 surface, application may provide only 1 buffers, but there is still only one single surface state for all of them. If IMC equal to 1, 2, 3 or 4, U and V have the pitch same as Y. And U and V will have different offset, each offset is block aligned.

DWord	Bit		Description						
0	31:29	Command [•]	Command Type						
		Default Val	ue:	3h PA	RALLEL_VIDEO_PIP	E			
		Format:		ОрСо	de				
	28:27	Pipeline	Pipeline						
		Default Val	ue:		2h MFX_COMMOI	N			
		Format:			OpCode				
	26:24	Opcode							
		Default Val	ue:	0h MF	X_COMMON_STAT	E			
		Format: OpCode							
	23:21	SubOpA							
		Default Value:				0h			
		Format:				OpCode			
	20:16	SubOpB							
		Default Value:				1h			
		Format:			OpCode				
	15:12	Reserved							
		Format:		Ν		1BZ			
	11:0	DWord Length							
		Format:	=n To	otal Le	ngth - 2				
		Value	N	ame		Description			
		4h	DWORD_COUNT_n [De	efault]		Excludes DWord (0,1)			



		1	MFX_SU	RFACE_STATE		
1	31:4	Reserved				
		Format:		MBZ		
		Surface Base Address	s is NOT used fo	or codec H/W. This filed is reserved for 3D surface sta	ite	
		compatibility. MFX pi	ipeline gets this	address from MFX_PIPE_BUF_ADDR_STATE for differ	ent	
		buffers.				
	3:0	Reserved				
		Format:		MBZ		
2	31:18	Height				
		Format:		U14-1 Height		
		This field specifies th	e height of the	Picture in units of pixels/residuals. For PLANAR surface	се	
		formats, this field ind	licates the heigl	ht of the Y (luma) plane. Note: Video Codecs must pro- ltiple of 2 MB rows for field picture/(C1 - mulitple of -	ogram 4 piyolo	
		for field pictureMPEG	52 - multiple of	2 MB rows for field picJPEG - mulitple of integral MC	U (8 or	
		16 pixels) per picture	•			
		Value	Name	Description		
		[0,16383]		representing heights [1,16384]		
		Programming Notes				
		• For AVC: For frame picture is a multiple of 16; for field picture is a multiple of 32				
		• For VC1: For progressive frames, the frame height and frame width is a multiple of 2 pixels. For interlaced frames, the frame height shall be a multiple of 4				
		pixels, and its width is a multiple of 2 pixels, based on a PLANAR_420 surface.				
		Video Codecs must program less than and equal to 4K.				
	17.4	Width				
	17.4	Format:		U14-1 Width		
		This field specifies th	e width of the F	Picture in units of pixels/residuals. For PLANAR surfac	е	
		formats, this field ind	licates the widtl	h of the Y (luma) plane.		
		Value	Name	Description		
		[0,16383]		representing widths [1,16384]		
			Dre	ogramming Notos		
		• The Width specified by this field multiplied by the pixel size in bytes must be less than or equal to the surface pitch (specified in bytes via the Surface Pitch				
		field).	quar to the sur	ace piter (specified in bytes via the burlace riter)		
		• Width (field v	alue + 1) must	be a multiple of 2 for PLANAR_420,		
		MFX HW doe	s not use this fi	eld, the picture width is read from IMG State		
		instead, beca	use this field m	ay not equal to the actual picture width. This field is		
		used by the K	IND to allocate			
		Video Codecs must	program less th	nan and equal to 4K.		



			MFX_SURFAC	CE_ST	ATE					
	3:2	Reserved								
		Format:			MBZ					
	1:0	Cr(V)/Cb(U) Pixel Offset V Direction								
		Format: U0.2 exactly as shown in the original spec								
		Specifies the distance to the U/V values with respect to the even numbered Y channels in the direction								
			Progra	mming	Notes					
		This field is	ignored for all formats except PL	ANAR_4	20_8					
3	31:28	Surface For	mat							
		Specifies the format of the surface. All of the Y and G channels will use table 0 and all of the Cr/Cb/R/B channels will use table 1.Usage: For 420 planar YUV surface, use 4; for monochrome surfaces, use 12. For monochrome surfaces, hardware ignores control fields for Chroma planes. This field must be set to 4 - PLANAR_420_8, or 12 - Y8_UNORMNot used for MFX, and is ignored. But for JPEG decoding, this field should be programmed to the same format as JPEG_PIC_STATE. For video codec, it should set to 4 always								
		Value	Name		Description					
		0	YCRCB_NORMAL							
		1	YCRCB_SWAPUVY							
		2	YCRCB_SWAPUV							
		3	YCRCB_SWAPY							
		4 PLANAR_420_8			(NV12, IMC1,2,3,4, YV12)					
		5	PLANAR_411_8		Deinterlace Only					
		6	PLANAR_422_8		Deinterlace Only					
		7	STMM_DN_STATISTICS		Deinterlace Only					
		8	R10G10B10A2_UNORM		Sample_8x8 Only					
		9	R8G8B8A8_UNORM		Sample_8x8 Only					
		10	R8B8_UNORM (CrCb		Sample_8x8 Only					
		11	R8_UNORM (Cr/Cb)		Sample_8x8 Only					
		12	Y8_UNORM		Sample_8x8 Only					
		13,15	Reserved							
	27	Interleave	Chroma							
		Format:		E	nable					
		This field in	dicates that the chroma fields are	interleav	yed in a single plane rather than stored as					
		two separat	e planes. This field is only used for est set to Enable to support interl	or PLANA eave U/V	R surface formats. For AVC/VC1/MPEG VLD					
		formats (inc	luding 4:2:0) - because JPEG doe	s not sup	port NV12. (This field is needed only if JPEG					
		will support	NV12; otherwise is ignored.)							
			Value		Name					
		1		Enable						
		0		Disable						



MFX_SURFACE_STATE

	Reserved							
	Format: MBZ							
25:22	Surface Object Control State (MEMORY_OBJECT_CONTROL_STATE)							
	This 4	-bit field is use	d in various state	comma	nds and indirect state objects.			
	Value	Name			Description			
	3	Reserved						
	2	Graphics Data Type (GFDT)	This field contains can also be set by field with the GFD Format = U1	the GFD1 the GTT. I from th	bit for this surface when writes occur. GF The effective GFDT is the logical OR of thi e GTT entry. This field is ignored for reads			
			Pro	grammin	g Notes			
	This fie entries	eld is ignored; H instead.	/W uses those valu	es progra	mmed in each of the Buf Address State			
21:20	Reserved							
	Forma	t:			MBZ			
19:3	Surface	e Pitch						
	Format: U17-1 pitch in Bytes							
	This field specifies the surface pitch in (#Bytes).							
	Value Name				Description			
	[0,204]	7]	[0,2047] to [1B, 2048B]					
			Pro	arammin	a Notes			
	For tile Pitch fe multip [128B,2	ed surfaces, the por Chroma is set le of 2 bytes for 256KB] = [1 tile,	Prop pitch must be a mu , this field must be linear surfaces. For 2048 tiles]	grammin Itiple of t a multipl Y-tiled s	g Notes he tile width (i.e.128 bytes aligned). If Hal e of two tile widths for tiled surfaces, or a urfaces: Range = [127, 524287] to			
2	For tile Pitch fe multip [128B,7 Half Pi	ed surfaces, the por Chroma is set le of 2 bytes for 256KB] = [1 tile, tch for Chroma	Prop pitch must be a mu , this field must be linear surfaces. For 2048 tiles]	grammin Itiple of t a multipl Y-tiled s	g Notes he tile width (i.e.128 bytes aligned). If Hal e of two tile widths for tiled surfaces, or a urfaces: Range = [127, 524287] to			
2	For tile Pitch fo multip [128B,7 Half Pi Forma	ed surfaces, the por Chroma is set le of 2 bytes for 256KB] = [1 tile, tch for Chroma t:	Prop pitch must be a mu , this field must be linear surfaces. For 2048 tiles]	grammin Itiple of t a multipl Y-tiled s	g Notes he tile width (i.e.128 bytes aligned). If Hal e of two tile widths for tiled surfaces, or a urfaces: Range = [127, 524287] to Enable			
2	For tile Pitch for multip [128B,7 Half Pi Forma (This fie equal to surface	ed surfaces, the p or Chroma is set le of 2 bytes for 256KB] = [1 tile, tch for Chroma t: eld must be set t o half the value formats. This fie	Prop pitch must be a mu , this field must be linear surfaces. For 2048 tiles] o Disable)This field specified in the Sur eld is igored by MF	rammin Itiple of t a multipl Y-tiled s V-tiled s	g Notes he tile width (i.e.128 bytes aligned). If Hal e of two tile widths for tiled surfaces, or a urfaces: Range = [127, 524287] to Enable s that the chroma plane(s) will use a pitch n field. This field is only used for PLANAR we support YV12)			
2	For tile Pitch for multip [128B,7 Half Pi Forma (This fie equal to surface	ed surfaces, the p or Chroma is set le of 2 bytes for 256KB] = [1 tile, tch for Chroma t: eld must be set t o half the value formats. This fie	Prop pitch must be a mu , this field must be linear surfaces. For 2048 tiles] o Disable)This field specified in the Sur eld is igored by MF	grammin Itiple of t a multipl Y-tiled s indicate face Pitch X (unless	g Notes he tile width (i.e.128 bytes aligned). If Hal e of two tile widths for tiled surfaces, or a urfaces: Range = [127, 524287] to Enable s that the chroma plane(s) will use a pitch n field. This field is only used for PLANAR we support YV12)			
2	For tile Pitch fo multip [128B,2 Half Pi Forma (This fie equal to surface Tiled S Forma	ed surfaces, the p or Chroma is set le of 2 bytes for 256KB] = [1 tile, tch for Chroma t: eld must be set t o half the value formats. This fie urface t:	Prop pitch must be a mu this field must be linear surfaces. For 2048 tiles] o Disable)This field specified in the Sur eld is igored by MF	rammin ltiple of t a multipl Y-tiled s indicate face Pitch X (unless Boo	g Notes he tile width (i.e.128 bytes aligned). If Hal e of two tile widths for tiled surfaces, or a urfaces: Range = [127, 524287] to Enable s that the chroma plane(s) will use a pitch n field. This field is only used for PLANAR we support YV12)			
2	For tile Pitch for multip [128B,7 Half Pi Forma (This fie surface Tiled S Forma (This fie ignored	ed surfaces, the p or Chroma is set le of 2 bytes for 256KB] = [1 tile, tch for Chroma t: eld must be set t formats. This fie urface t: eld must be set t d by MFX	Prop pitch must be a mu ; this field must be linear surfaces. For 2048 tiles] o Disable)This field specified in the Sur eld is igored by MF	rammin Itiple of t a multipl Y-tiled s indicate face Pitch X (unless Boo field spec	g Notes he tile width (i.e.128 bytes aligned). If Hal e of two tile widths for tiled surfaces, or a urfaces: Range = [127, 524287] to Enable s that the chroma plane(s) will use a pitch h field. This field is only used for PLANAR we support YV12) blean ifies whether the surface is tiled. This field			
2	For tile Pitch fe multip [128B,7 Half Pi Forma (This fie equal to surface Tiled S Forma (This fie ignored	ed surfaces, the p or Chroma is set le of 2 bytes for 256KB] = [1 tile, tch for Chroma t: eld must be set t formats. This fie urface t: eld must be set t d by MFX Value	Prop pitch must be a mu t, this field must be linear surfaces. For 2048 tiles] o Disable)This field specified in the Sur eld is igored by MF o TRUE: Tiled)This Name	rammin Itiple of t a multipl Y-tiled s indicate face Pitch X (unless Boo field spec	g Notes he tile width (i.e.128 bytes aligned). If Hal e of two tile widths for tiled surfaces, or a urfaces: Range = [127, 524287] to Enable s that the chroma plane(s) will use a pitch n field. This field is only used for PLANAR we support YV12) Delean ifies whether the surface is tiled. This field			
2	For tile Pitch fo multip [128B,7 Half Pi Forma (This fie equal to surface Tiled S Forma (This fie ignorec	ed surfaces, the p or Chroma is set le of 2 bytes for 256KB] = [1 tile, tch for Chroma t: eld must be set t to half the value formats. This fie urface t: eld must be set t d by MFX Value	Prop pitch must be a mu ; this field must be linear surfaces. For 2048 tiles] o Disable)This field specified in the Sur eld is igored by MF o TRUE: Tiled)This Name False	rammin ltiple of t a multipl Y-tiled s l indicate face Pitch X (unless leld spec	g Notes he tile width (i.e.128 bytes aligned). If Hal e of two tile widths for tiled surfaces, or a urfaces: Range = [127, 524287] to Enable s that the chroma plane(s) will use a pitch h field. This field is only used for PLANAR we support YV12) blean ifies whether the surface is tiled. This field Description Linear			



MFX_SURFACE_STATE

		F					
		Programming Notes					
Linear surfaces can be mapped to Main Memory (uncached) or System Memory (ca snooped). Tiled surfaces can only be mapped to Main Memory. The corresponding must be invalidated before a previously accessed surface is accessed again with an of this bit.							
	0	Tile Walk					
		Format:	3D	Tilewalk			
		(This field must be set to 1: TILEWALK_YMAJOR)This field specifies the type of memory tiling (XMajor or YMajor) employed to tile this surface. See Memory Interface Functions for details on memory tiling and restrictions. This field is ignored when the surface is linear. This field is ignored by MFX. Internally H/W is always treated this set to 1 for all video codec and for JPEG.					
		Value	Name		Description		
		0h	XMAJOR	TILEWALK_X	MAJOR		
		1h	YMAJOR	TILEWALK_Y	MAJOR		
			Prog	ramming Note	es		
		The corresponding accessed again wi	g cache(s) must be invali th an altered state of this	dated before a s bit	previously accessed surface is		
4	31	Reserved					
		Format:			MBZ		
	30:16	X Offset for U(Cb))				
		Format:	U15 Pixel	Offset			
		This field specifies the horizontal offset in pixels from the Surface Base Address to the start (origin) of the U(Cb) plane or the interleaved UV plane if Interleave Chroma is enabled. This field is only used for PLANAR surface formats. This field must be set to zero. X Offset for U(Cb) in pixel (This field must be zero for NV12 and IMC 1 and 3)					
		(origin) of the U(Ct is only used for PL/ (This field must be	o) plane or the interleave ANAR surface formats. The zero for NV12 and IMC	d UV plane if In his field must b L and 3)	nterleave Chroma is enabled. This field e set to zero. X Offset for U(Cb) in pixel		
		(origin) of the U(Ck is only used for PL/ (This field must be	o) plane or the interleave ANAR surface formats. Th zero for NV12 and IMC : Prog	d UV plane if In his field must b L and 3) ramming Note	nterleave Chroma is enabled. This field e set to zero. X Offset for U(Cb) in pixel es		
		(origin) of the U(Ck is only used for PL/ (This field must be For PLANAR_420 a	o) plane or the interleave ANAR surface formats. The zero for NV12 and IMC : Prog and PLANAR_422 surface	d UV plane if In his field must b L and 3) ramming Note formats, this f	nterleave Chroma is enabled. This field e set to zero. X Offset for U(Cb) in pixel es ield must be zero.		
	15	(origin) of the U(Ck is only used for PL/ (This field must be For PLANAR_420 a Reserved	o) plane or the interleave ANAR surface formats. The zero for NV12 and IMC : Prog and PLANAR_422 surface	d UV plane if In his field must b L and 3) ramming Note formats, this f	nterleave Chroma is enabled. This field e set to zero. X Offset for U(Cb) in pixel es ield must be zero.		
	15	(origin) of the U(Ck is only used for PL/ (This field must be For PLANAR_420 a Reserved Format:	o) plane or the interleave ANAR surface formats. Th zero for NV12 and IMC : Prog and PLANAR_422 surface	d UV plane if In his field must b L and 3) ramming Note formats, this f	nterleave Chroma is enabled. This field e set to zero. X Offset for U(Cb) in pixel es ield must be zero. MBZ		
	15	(origin) of the U(Ck is only used for PL/ (This field must be For PLANAR_420 a Reserved Format: Y Offset for U(Cb)	o) plane or the interleave ANAR surface formats. The zero for NV12 and IMC : Prog and PLANAR_422 surface	d UV plane if In his field must b L and 3) ramming Note formats, this f	nterleave Chroma is enabled. This field e set to zero. X Offset for U(Cb) in pixel es ield must be zero. MBZ		
	15	(origin) of the U(Ck is only used for PL/ (This field must be For PLANAR_420 a Reserved Format: Y Offset for U(Cb) Format:	o) plane or the interleave ANAR surface formats. The zero for NV12 and IMC : Prog and PLANAR_422 surface U15 Pixel Row	d UV plane if In his field must b L and 3) ramming Note formats, this f Offset	nterleave Chroma is enabled. This field e set to zero. X Offset for U(Cb) in pixel es ield must be zero. MBZ		
	15	(origin) of the U(Ck is only used for PL/ (This field must be For PLANAR_420 a Reserved Format: Y Offset for U(Cb) Format: This field specifies of the U(Cb) plane used for PLANAR s	b) plane or the interleave ANAR surface formats. The zero for NV12 and IMC : Prog and PLANAR_422 surface U15 Pixel Row the veritical offset in row or the interleaved UV pla surface formats.	d UV plane if In his field must b L and 3) ramming Note formats, this f Offset offset s from the Sur ane if Interleave	Interleave Chroma is enabled. This field Image: Set to zero. X Offset for U(Cb) in pixel Image: Set to zero.		
	15	(origin) of the U(Ck is only used for PL/ (This field must be For PLANAR_420 a Reserved Format: Y Offset for U(Cb) Format: This field specifies of the U(Cb) plane used for PLANAR s	b) plane or the interleave ANAR surface formats. The zero for NV12 and IMC : Prog and PLANAR_422 surface U15 Pixel Row the veritical offset in row or the interleaved UV plass surface formats. Prog	d UV plane if In his field must b L and 3) ramming Note formats, this f Offset s from the Sur ane if Interleave ramming Note	Interleave Chroma is enabled. This field are set to zero. X Offset for U(Cb) in pixel Image: Set to zero. Image: Set to zero. Image: MBZ Image: Set to zero. Image: Set to		
	15 14:0	(origin) of the U(Ck is only used for PL/ (This field must be For PLANAR_420 a Reserved Format: Y Offset for U(Cb) Format: This field specifies of the U(Cb) plane used for PLANAR s For PLANAR_420 a multiple MBs. For	b) plane or the interleave ANAR surface formats. The zero for NV12 and IMC : Prog and PLANAR_422 surface U15 Pixel Row the veritical offset in row or the interleaved UV plas surface formats. Prog and PLANAR_422 surface JPEG, this field must be a	d UV plane if In his field must b L and 3) ramming Note formats, this f Offset s from the Sur ane if Interleave ramming Note formats, this f a multiple of 16	hterleave Chroma is enabled. This field e set to zero. X Offset for U(Cb) in pixel es ield must be zero. MBZ face Base Address to the start (origin) e Chroma is enabled. This field is only es ield must be multiple of 16 pixels - i.e. 5 pixels.		
5	15 14:0 31:29	Inis field specifies (origin) of the U(Ck is only used for PL/ (This field must be For PLANAR_420 a Reserved Format: Y Offset for U(Cb) Format: This field specifies of the U(Cb) plane used for PLANAR_s For PLANAR_420 a multiple MBs. For Reserved	b) plane or the interleave ANAR surface formats. The zero for NV12 and IMC : Prog and PLANAR_422 surface U15 Pixel Row the veritical offset in row or the interleaved UV pla surface formats. Prog and PLANAR_422 surface JPEG, this field must be a	d UV plane if In his field must b L and 3) ramming Note formats, this f Offset s from the Sur ane if Interleave ramming Note formats, this f a multiple of 16	Anterleave Chroma is enabled. This field are set to zero. X Offset for U(Cb) in pixel and the set to zero. In the set of		



	MFX_SURFACE_STATE					
28	X Offset for V(Cr)					
	Format: U13 Offset in Pixels					
	This field must be zero for NV12 and IMC 1 and 3					
	This field specifies the horizontal offset in pixels from the Surface Base Address to the start (origin) of the V(Cr) plane. This field is only used for PLANAR surface formats with Interleave Chroma disabled.					
	Programming Notes For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of					
	pixels.					
1	5:0 Y Offset for V(Cr)					
	Format: U16 Row Offset in Pixels					
	This field specifies the veritical offset in rows from the Surface Base Address to the start (origin) of the V(Cr) plane. This field is only used for PLANAR surface formats with Interleave Chroma disabled. This field is ignored by all video codec, only used by JPEG.					
	Programming Notes					
	For PLANAR_420 surface formats, this field must be multiple of 16 pixels - i.e. multiple MBs. For JPEG, this field must be a multiple of 16 pixels.					



MFX VC1	DIRECTMODE STATE

Source: VideoCS

Length Bias:

This is a picture level command and should be issued only once, even for a multi-slices picture. This command is only valid in the VC1 decoding in VLD modes. There is only one DMV buffer for read (when processing a B-picture) and one for write (when processing a P-Picture). Each DMV record is 64 Bytes per MB, to store the top and bottom field MVs (32-bit MVx,y each). Note that if there is a I picture before a B picture the DmvSurfaceValid state in MFX_VC1_PIC_STATE Command will NOT be set and zero's DMV's will be assumed while decoding the B picture. That is, there is no explicit DMV buffer for an I-picture.

DWord	Bit	Description						
0	31:29	Command Type						
		Default Value: 3h PARALLEL_			L_VID	VIDEO_PIPE		
	Format:			OpCode				
	28:27	Pipeline						
	Default Value: 2h MFX_VC1_DIRECTMODE_STATE				TATE			
		Format:	Ор	Code				
	26:24	Media Command	l Opcode					
		Default Value:					2h VC1	
		Format:					OpCode	
	23:21	SubOpcode A						
		Default Value:			Oł	n Com	imon	
		Format:			0	OpCode		
	20:16	SubOpcode B						
		Default Value:				2h MEDIA_		
		Format:				OpCode		
	15:12	Reserved						1
		Format:				I	MBZ	
	11:0	DWord Length						
		Default Value:		0001h Exclude	es DW	ord (0	0,1)	
		Format: =n Total Length			th - 2	1 - 2		
1	31:6	1:6 Direct MV Write Buffer Base Address for the Current Picture This field provides the base address of the DMV write buffer to store the motion vectors dec in the current picture. It is a private buffer used by the MPR hardware only. Its content is not accessed by software. This buffer must be 64-byte cacheline aligned. The write buffer size is 557,056 bytes for 1 frame. Scalable with frame height, but do not scale with frame width as t hardware assumes frame width (in MBs) fixed at 128 (smallest power of 2 value larger than 1 1920x1088 screen resolution). This field is only valid for a P picture					otion vectors decoded ts content is not ite buffer size is frame width as the lue larger than 120 -	
	5:4	Direct MV Write	Buffer Base Ad	ddress - Arbiti	ation	Prior	ity Control	
		Format:	U2 E	numerated Ty	be			
		This field controls	the priority of a	arbitration use	d in th	e GAC	C/GAM pipeline f	or this surface. 1
		Value		Nar	ne			



MFX_VC1_DIRECTMODE_STATE

		00b		Highest priority			
		01b		Second highest priority			
		10b		Third highest priority			
		11b		Lowest priority			
	3 Reserved						
	2	Direct MV Write Buffer Base Address - Graphics Data Type (GFDT) for the Current Picture					
		Default Val	lue:		0h		
		Format:			U1		
		This field co	ontains	the GFDT bit for this surface when writes occur. GFDT can	also be set by the		
		GTT. The ef	fective	GFDT is the logical OR of this field with the GFDT from the	e GTT entry. This field		
		is ignored to	or read	15.			
	1:0	Direct MV	Write	Buffer Base Address - Cacheability Control for the Curr	ent Picture		
		Format:		U2 Enumerated Type			
		This field co	ontrols	cacheability.			
		Value		Name			
		00b	use c	acheability control bits from GTT entry			
		01b	data	s not cached			
		11b	data	s cached			
2	31:6	Direct MV	Read	Buffer Base Address for the Reference Picture			
		This field pr	rovides	the base address of the DMV buffer for reference picture.	It is a private buffer		
		used by the	MPR	hardware only. Its content is not accessed by software. All	these buffers must		
		be 64-byte	cachel	ine aligned. This field is only valid for a B picture.			
	5:4	Direct MV	Read	Buffer - Arbitration Priority Control			
		Format:	ntrolo	U2 Enumerated Type	or this surface		
			ontrois	the priority of arbitration used in the GAC/GAM pipeline for	or this surface.		
		Value	•	Name			
		000		Energy designed and an and a second s			
		010					
		100		I nird nignest priority			
		110		Lowest priority			
	3	Reserved					
	2	Direct MV	Read	Buffer - Graphics Data Type (GFDT) for the Reference P	icture		
		Format:					
		This field co	ontains	the GFDT bit for this surface when writes occur. GFDT can	also be set by the		
		is janored f	or read	Is \	e di l'entry. This heid		
		is ignored is	2				
	1:0	Direct MV	Read	Buffer - Cacheability Control for the Reference Picture			
		Format:		U2 Enumerated Type			



MFX_VC1_DIRECTMODE_STATE					
	This field co	ontrols cacheability.			
	Value	Name			
	00b	use cacheability control bits from GTT entry			
	01b	data is not cached			
	11b	data is cached			



VideoCS

2

MFX_VC1_PRED_PIPE_STATE

Source:

Length Bias:

This command is used to set the operating states of the MFD Engine beyond the BSD unit. It is used with both VC1 Long and Short format.Driver is responsible to take the intensity compensation enable signal, the LumScale and the LumShift provided from the DXVA2 VC1 interface, and maintain a history of these values for reference pictures. Together with these three parameters specified for the current picture being decoded, driver will derive and supply the above sets of LumScaleX, LumShiftX and intensity compensation enable (single or double, forward or backward) signals. H/W is responsible to take these state values, and use them to build the lookup table (including the derivation of iScale and iShift) for remapping the reference frame pixels, as well as perfoming the actual pixel remapping calculations/process.

DWord	Bit	Description						
0	31:29	Command Type						
		Default Value: 3h PARALLEL_VIDEO_PIPE						
		Format: OpCode						
	28:27	Pipeline						
		Default Value:	2h MFX_VC1_PRED_PIPE_STATE					
		Format:	OpCod	le				
	26:24	Media Command Opcode						
		Default Value:			2h VC1_COMM	ON		
		Format:			OpCode			
	23:21	SubOpcode A						
		Default Value:				0h		
		Format:				ОрСо	de	
	20:16	SubOpcode B						
		Default Value:				1h		
		Format:				ОрСо	de	
	15:12	Reserved						
		Format:				MBZ		
	11:0	DWord Length						
		Default Value:	0004	h E	xcludes DWord (ludes DWord (0,1)		
		Format:	=n T	ota	l Length - 2			
1	31:16	Reserved						
		Format:				MBZ		
	15:14	vin_intensitycomp_Double_FV	VDen					
		Format:					U2	
		for forward reference picture or	nly, to e	enat	ole top or/and bo	ottom o	f the reference field enable for	
		single compensation. For frame,	, may o widod k	only	need one bit.	na and	chart VC1 interface format	
		And is derived from the intensit	v comp	by c bens	sation enable flag	ng anu a wBitst	reamPCFelement and	
		wBitstreamFcodes parameters p	provideo	d by	y the DXVA2 VC1	1 interfa	ce to the driver for each	



MFX_VC1_PRED_PIPE_STATE		
	current picture.	
13:12	vin_intensitycomp_Double_BWDen	
	Format:	U2
	for backward reference picture only, no double for backward	ard reference.
	This field is maintained and provided by driver for both long and short VC1 interface format. And is derived from the intensity compensation enable flag, wBitstreamPCEelement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.	
11:10	vin_intensitycomp_Single_FWDen	
	Format:	U2
	single compensation. For frame, may only need one bit. This field is maintained and provided by driver for both long and short VC1 interface And is derived from the intensity compensation enable flag, wBitstreamPCEelement ar wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for current picture.	
9:8	vin_intensitycomp_Single_BWDen	
	Format:	U2
	This field is maintained and provided by driver for both long and short VC1 interface format. And is derived from the intensity compensation enable flag, wBitstreamPCEelement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.	
7:4	Reference Frame Boundary Replication Mode	
	Format:	U4
	This is a bit field with each bit indicating the correspondin Bit 11: reference 3 Bit 10: reference 2 Bit 9: reference 1 Bit 8: reference 0	g picture's boundary replication mode
	0 = progressive frame replication 1 = interlace frame replication	
	This field is maintained and provided by driver for both long and short VC1 interface format.	
3:0	Reserved	
	Format:	MBZ
31:30	Reserved	1
	Format:	MBZ


29:24	LumShift2- single - FWD					
	Format:	U6				
	This field is maintained and provid And is derived from the intensity of wBitstreamFcodes parameters pro current picture.	This field is maintained and provided by driver for both long and short VC1 interface format. And is derived from the intensity compensation enable flag, wBitstreamPCEelement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.				
23:22	Reserved					
	Format:	MBZ				
21:16	LumShift1 - single - FWD					
	Format:	U6				
	And is derived from the intensity of wBitstreamFcodes parameters pro- current picture.	compensation enable flag, wBitstreamPCEelement and swided by the DXVA2 VC1 interface to the driver for each				
15:14	Reserved					
	Format:	MBZ				
13:8	LumScale2 - single - FWD					
	Format:	U6				
	And is derived from the intensity of wBitstreamFcodes parameters procurrent picture.	compensation enable flag, wBitstreamPCEelement and svided by the DXVA2 VC1 interface to the driver for each				
7:6	Reserved					
	Format:	MBZ				
5:0	LumScale1 - Single - FWD					
	Format:	U6				
	This field is maintained and provid And is derived from the intensity of wBitstreamFcodes parameters pro current picture.	ded by driver for both long and short VC1 interface format compensation enable flag, wBitstreamPCEelement and ovided by the DXVA2 VC1 interface to the driver for each				
	Reserved					
31:30		MBZ				
31:30	Format:					
31:30	LumShift2- double - FWD					
31:30 29:24	LumShift2- double - FWD Format:	U6				



		MFX_VC1_PRED_PIPE_S	TAT	E				
	23:22	Reserved						
		Format:	MBZ					
	21:16	LumShift1 - double -FWD						
		Format:		U6				
		is field is maintained andprovided by driver for both long and short VC1 interface format. And derived from the intensity compensation enable flag, wBitstreamPCEelement and BitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each rrent picture.						
	15:14	Reserved						
	Format: MBZ							
	13:8	LumScale2 - double - FWD	•					
		Format:		U6				
		hort VC1 interface format. And nPCEelement and ice to the driver for each						
	7:6	Reserved						
		Format:	MBZ					
	5:0	LumScale1 - double - FWD						
		Format:		U6				
		This field is maintained andprovided by driver for both long and short VC1 interface format. And is derived from the intensity compensation enable flag, wBitstreamPCEelement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.						
4	31:30	Reserved						
		Format:	MBZ					
	29:24	LumShift2- single - BWD						
		Format:		U6				
		This field is maintained andprovided by driver for both long and short VC1 interface format. And is derived from the intensity compensation enable flag, wBitstreamPCEelement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.						
	23:22	Reserved						
		Format:	MBZ					
	21:16	LumShift1 - single - BWD						
		Format:		U6				
		This field is maintained andprovided by driver for both lor is derived from the intensity compensation enable flag, wh	ng and s Bitstrean	hort VC1 interface format. And nPCEelement and				



		MFX_VC1_PRED_PIPE_S	TATE					
		wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.						
	15:14 Reserved							
		Format: MBZ						
	13:8	LumScale2 - single - BWD						
		Format:		U6				
		This field is maintained andprovided by driver for both lon is derived from the intensity compensation enable flag, we wBitstreamFcodes parameters provided by the DXVA2 VC2 current picture.	his field is maintained andprovided by driver for both long and short VC1 interface format. And derived from the intensity compensation enable flag, wBitstreamPCEelement and BitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each urrent picture.					
	7:6	Reserved						
		Format:	MBZ					
	5:0	LumScale1 - Single - BWD						
		Format:		U6				
		This field is maintained and provided by driver for both lon is derived from the intensity compensation enable flag, we wBitstreamFcodes parameters provided by the DXVA2 VC2 current picture.	ig and sh Bitstream 1 interfac	PCEelement and PCEelement and the driver for each				
5	31:30	Reserved		1				
		Format:	MBZ					
	29:24	LumShift2- double - BWD		1				
		Format:		U6				
	This field is maintained andprovided by driver for both long and short VC1 interface form is derived from the intensity compensation enable flag, wBitstreamPCEelement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.							
	23:22	Reserved						
		Format:	MBZ					
	21:16	LumShift1 - double -BWD						
		Format:		U6				
		This field is maintained andprovided by driver for both long and short VC1 interface format. An is derived from the intensity compensation enable flag, wBitstreamPCEelement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.						
	15:14	Reserved						
		Format:	MBZ					
	13:8	LumScale2 - double - BWD						



MFX_VC1_PRED_PIPE_STATE

	Format: This field is maintained andprovided by driver for both lon is derived from the intensity compensation enable flag, wB wBitstreamFcodes parameters provided by the DXVA2 VC1 current picture.	g and sl Bitstrean L interfa	U6 hort VC1 interface format. And hPCEelement and ce to the driver for each		
7:6	7:6 Reserved				
	Format: MBZ				
5:0	LumScale1 - double - BWD				
	Format:		U6		
This field is maintained andprovided by driver for both long and short VC1 inter is derived from the intensity compensation enable flag, wBitstreamPCEelement a wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the drive current picture.					



MFX_WAIT

Source: VideoCS

1

Length Bias:

This command can be considered the same as an MI_NOOP except that the command parser will not parse the next command until the following happens

- AVC or VC1 BSD mode: The command will stall the parser until completion of the BSD object
- **IT, encoder, and MPEG2 BSD mode:** The command will stall the parser until the object package is sent down the pipelineThis command should be used to ensure the preemption enable window occurs during the time the object command is being executed down the pipeline.

DWord	Bit	Description				
0	31:29	Command Type				
		Default Value: 03h PARALLEL_VIDEO_PIPE				
		Format:	OpCode			
	28:27	Command Subtype	d Subtype			
		Default Value:	01h MFX_SINGLE_DW			_DW
		Format:	OpC	ode		
	26:16	Sub-Opcode				
		Default Value:			0h MFX	_WAIT
		Format:			OpCode	2
	15:10	Reserved				
		Format:				MBZ
	9	Reserved				
8 MFX Sync Control Flag If set, VCS will stall the parser until all prior MF pipeline			FX objec	ts are completed down the MFX		
	7:6	Reserved				
		Format:				MBZ
	5:0	DWord Length	-			
Default Value: 0h Excludes DWord (0,1)			(0,1)			
		Format: =n				
		Total Length - 2				



MI	ARB	CHFCK

Source: BlitterCS

Length Bias:

The MI_ARB_CHECK is used to check for a change in arbitration. If executed as part of a Ring Buffer the command checks the UHPTR valid bit and if set the head of the ring will jump to the value of the head pointer programmed in the UHPTR.

Programming Notes						
This instruction car	This instruction cannot be placed in a batch buffer.					
DWord	Bit	Bit Description				
0	31:29	Command Type				
		Default Value:	0h MI_INSTRUCTION			
		Format:	ormat: OpCode			
	28:23	MI Command Opcode				
		Default Value:	05h MI_ARB_	СНЕСК		
		Format:	OpCode			
	22:0	Reserved				
		Format:		MBZ		



MI_ARB_CHECK

Source: RenderCS

1

Length Bias:

The MI_ARB_CHECK instruction is used to check the ring buffer double buffered head pointer (register UHPTR). This instruction can be used to pre-empt the current execution of the ring buffer. Note that the valid bit in the updated head pointer register needs to be set for the command streamer to be pre-empted.

Programming Notes

- The current head pointer is loaded with the updated head pointer register independent of the location of the updated head.
- If the current head pointer and the updated head pointer register are equal, hardware will automatically reset the valid bit corresponding to the UHPTR.
- For pre-emption, the wrap count in the ring buffer head register is no longer maintained by hardware. The hardware updates the wrap count to the value in the UHPTR register.

This instruction can be in either a ring buffer or batch buffer.

DWord	Bit	Description				
0	31:29	Command Type				
		Default Value: 0h MI_COMMAND				
		Format:	OpCode			
	28:23	MI Command Opcode				
		Default Value:	05h MI_ARB_CHECK			
		Format:	OpCode			
	22:0	Reserved				
		Format:	MBZ			



MI	ΔRB	CHFC	Κ

Source: VideoCS

Length Bias:

The MI_ARB_CHECK is used to check for a change in arbitration. If executed as part of a Ring Buffer the command checks the UHPTR valid bit and if set the head of the ring will jump to the value of the head pointer programmed in the UHPTR.

Programming Notes						
This instruction ca	This instruction cannot be placed in a batch buffer.					
DWord	Bit	Description				
0	31:29	MI Instruction Type				
		Default Value:	t Value: 0h MI_INSTRUCTION			
		Format: OpCode				
	28:23	MI Instruction Opcode				
		Default Value:		05h MI_ARB_	СНЕСК	
		Format:		OpCode		
	22:0	Reserved				
		Format:			MBZ	



MI_ARB_ON_OFF

Source: RenderCS

1

Length Bias:

The MI_ARB_ON_OFF instruction is used to mask/differ the below asynchronous events when arbitration is disabled: • PSMI Context Switch Request • Sync Flush • Power FLush Block This command should always be used as an off-on pair around the sequence of instructions to be protected from above mentioned asynchronous events.

DWord	Bit	Description			
0	31:29	Command Type			
		Default Value: 0h MI_COMMAND			
		Format: OpCode			
	28:23	MI Command Opcode			
		Default Value: 08h MI_ARB_ON_OFF			N_OFF
		Format: OpCode			
	22:1	Reserved			
		Format:			MBZ
	0	Arbitration Enable			
		Format: Enable			le
		This field enables or disables arbitration in HW.			
		Value Name			
		0h Disabled			
		1h	Enabled		



MI_ARB_ON_OFF

Source: VideoCS

Length Bias:

The MI_ARB_ON_OFF instruction is used to mask/differ the below asynchronous events when arbitration is disabled: • PSMI Context Switch Request • Sync Flush • Power FLush Block This command should always be used as an off-on pair around the sequence of instructions to be protected from above mentioned asynchronous events.

DWord	Bit	Description			
0	31:29	Command Type			
		Default Value: 0h MI_COMMAND			MAND
	28:23	MI Command Opcode			
		Default Value:	08h MI_A	RB_O	N_OFF
	22:1	Reserved			
		Format: MBZ			MBZ
	0	Arbitration Enable			
		Format: Enable			le
		This field enables or disables arbitration in HW.			
		Value Name			
		0h Disabled			
		1h	Enabled		



MI_BATCH_BUFFER_END

Source:	BlitterC	S	5					
Length Bias:	1							
The MI_BATCH_BUFFER_END command is used to terminate the execution of commands stored in a batch buffe initiated using a MI_BATCH_BUFFER_START command.								
DWord	Bit	Description						
0	31:29	Command Type		1				
		Default Value:		0h MI_COMN	IAND			
	28:23	MI Command Opcode						
		Default Value: 0Ah MI_ BATCH_BUFFER_END						
	22:0	Reserved						
		Format:			MBZ			



MI_BATCH_BUFFER_END Source: RenderCS Length Bias: 1 The MI_BATCH_BUFFER_END command is used to terminate the execution of commands stored in a batch buffer initiated using a MI_BATCH_BUFFER_START command. DWord Description Bit 0 31:29 Command Type Default Value: 0h MI_COMMAND Format: OpCode 28:23 MI Command Opcode Default Value: 0Ah MI_ BATCH_BUFFER_END Format: OpCode Reserved 22:0 Format: MBZ



MI_BATCH_BUFFER_END

Source:	Video	CS			
Length Bias:	1				
The MI_BATCH_BUFFER_END command is used to terminate the execution of commands stored in a batch b initiated using a MI_BATCH_BUFFER_START command.					
DWord	Bit	Description			
0	31:29	Command Type	Command Type		
		Default Value:		0h MI_COMMAND	
		Format:		OpCode	
	28:23	MI Command Opcode			
		Default Value:	0Ah MI_	I_BATCH+_BUFFER_END	
		Format: OpCode			
	22:0	Reserved			
		Format:			MBZ



MI_BATCH_BUFFER_START

Source: BlitterCS

Length Bias:

The MI_BATCH_BUFFER_START command is used to initiate the execution of commands stored in a batch buffer. For restrictions on the location of batch buffers, see Batch Buffers in the Device Programming Interface chapter of MI Functions. The batch buffer can be specified as secure or non-secure, determining the operations considered valid when initiated from within the buffer and any attached (chained) batch buffers. See Batch Buffer Protection in the Device Programming Interface chapter of MI Functions.

Programming Notes

• A batch buffer initiated with this command must end either with a MI_BATCH_BUFFER_END command or by chaining to another batch buffer with an MI_BATCH_BUFFER_START command.

DWord	Bit	Description							
0	31:29	Commar	Command Type						
		Default V	Value:			MAND			
		Format:				OpCode			
	28:23	MI Com	mand O	pcode					
		Default	Value:		31h MI	_BATCH_BUFFE	R_START		
		Format:			OpCod	e			
	22	Reserved	3						
		Format:					MBZ		
	21:9	Reserved	3						
		Format:					MBZ		
	8	Address	Space I	ndicato	r				
		Format:	-		MI_BufferSe	curityType			
				1					
		Value	Name			Desc	ription		
		0h	GGTT	This ba	atch buffer is	secure and will	l be acce	essed via the	GGTT.
					Dr	aramming No	toc		
		This field	d must h	a '0' un	less the Per-	Process GTT En	able is '1	1	
	7.0			e o un				-	
	7:0	Eormat:	.engtn					-n	
		Total - Bi	as					-11	
		Value Name							
		0h	0h Excludes DWord (0,1) [Default]						
1	31:2	Batch Buffer Start Address							
		Format:		Graph	icsAddress[31:2]BatchBuffe	r		
		This field	specifie	s Bits 31	L:2 of the sta	rting address o	f the bat	ch buffer.	



MI_BATCH_BUFFER_START								
1:0	Reserved							
	Format:	MBZ						



Source: VideoCS

Length Bias:

The MI_BATCH_BUFFER_START command is used to initiate the execution of commands stored in a batch buffer. For restrictions on the location of batch buffers, see Batch Buffers in the Device Programming Interface chapter of MI Functions. The batch buffer can be specified as secure or non-secure, determining the operations considered valid when initiated from within the buffer and any attached (chained) batch buffers. See Batch Buffer Protection in the Device Programming Interface chapter of MI Functions.

DWord	Bit		Description							
0	31:29	Comma	and Type							
		Default	: Value:			0h MI_COMMA	ND			
		Format	•			OpCode				
	28:23	MI Con	nmand Opcod	e						
		Default	: Value:		31h MI_BA	ATCH_BUFFER_ST	TART			
		Format			OpCode					
	22	2nd Lev	el Batch Buff	er						
		The con head ac chaining is no sta storage (traditio	command streamer contains 3 storage elements; 1 for the ring head address, 1 for the batch ad address, and 1 for the 2nd level batch head address. When performing batch buffer ining, hardware simply updates the head pointer of the 1st level batch address storage. There to stack in hardware. When this bit is set, hardware uses the 2nd level batch head address rage element. Upon MI_BATCH_BUFFER_END, it will automatically return to the 1st additional) level batch buffer address. this allows hardware to mimic a simple 3 level stack.							
Value Name Des					Desc	Description				
		0h	1st level batch	Place th storage	Place the batch buffer address in the 1st (traditional) level batch address storage element					
		1h	2nd level batch	Place th element	Place the batch buffer address in the 2nd level batch address storage element					
		Programming Notes								
		•	2nd level batch	n buffer o	haining is r	ot supported.				
	21:10	Reserved								
		Format	•				MBZ			
	9	Reserve	ed							
	8	Address Space Indicator								
					Descr	iption				
		This field must be 0 unless the Per-Process GTT Enable is 1.								
		Value	e Name			D	escription			
		0	GGTT space	e Tł	nis batch bu	Iffer will be acces	ssed via the GGTT.			
		1	PPGTT	TI	nis batch bu	Iffer will be acces	ssed via the PPGTT.			



			MI_BA	TCH_BUFFER_ST	ART					
	7:0	DWord Length								
		Format:		=n Total Length - 2						
		Value		Name						
		0h	Excludes DV	Vord (0,1) [Default]						
1	31:2	Batch Buffer	Start Addres	S						
		Format:		GraphicsAddress[31:2]						
		Programming Notes A batch buffer initiated with this command must end either with a MI_BATCH_BUFFER_END command or by chaining to another batch buffer with an MI_BATCH_BUFFER_START command. The selection of PPGTT vs. GGTT for the batch buffer is determined by the Buffer Securit								
	1:0	Reserved Format: MBZ								



MI_BATCH_BUFFER_START

Source: RenderCS

Length Bias:

The MI_BATCH_BUFFER_START command is used to initiate the execution of commands stored in a batch buffer. For restrictions on the location of batch buffers, see Batch Buffers in the Device Programming Interface chapter of MI Functions.

Programming Notes

It is essential that the address location beyond the current page be populated inside the GTT. HW performs over-fetch of the command addresses and any over-fetch requires a valid TLB entry. A single extra page beyond the batch buffer is sufficient. Prior to sending batch buffer start command with clear command buffer enable set, software has to ensure pipe is flushed explicitly by sending MI_FLUSH.

DWord	Bit	Description						
0	31:29	Command Type						
		Default Value:	0h MI_COMMAND					
		Format:		OpCode				
	28:23	MI Command Opcode						
		Default Value:	31h MI_BA	ATCH_BUF	FER_S	TART		
		Format:	OpCode					
	22	Reserved						
		Format:				MBZ		
	21:17	Reserved						
		Format:				MBZ		
	16	Reserved						
		Format:				MBZ		
	15	Reserved						
		Format:				MBZ		
	14	Reserved						
		Format:				MBZ		
	13	Reserved						
		Format:				MBZ		
	12	Reserved						
	11	Clear Command Buffer Enable	e					
		Format:			Enable	2		
		This batch buffer needs to be preceded by a MI_FLUSH command or PIPE_CONTROL with CS Stall set.						
	10	Reserved						
		Format:				MBZ		
	9	Reserved						



MI_BATCH_BUFFER_START

		Format:				MBZ						
	8	Address S	oace Indica	tor								
			Description									
		SW must e as the initi Buffer has buffers (no Not comp	SW must ensure the "Address Space Indicator" of the chained batch buffer to be same as the initial batch buffer. Ex: If the MI_BATCH_BUFFER_START executed from Ring Buffer has "Address Space Indicator" as "PPGTT" then all subsequent chained batch buffers (not second level Batch Buffers) must be in "PPGTT". Not complying to above programming will result in unknown behavior of HW.									
		This field r	nust be '0' ı	unless the Per-	-Process GTT Enable is	5 '1'						
		Value	Name		Dese	cription						
		0h	GGTT	This batch bu	uffer will be accessed v	via the GGTT.						
		1h	PPGTT	This batch bu	uffer will be accessed v	via the PPGTT.						
	7:0	DWord Le	ngth									
		Default Va	lue:		0h Excludes DWord ((0,1)						
		Format:			=n Total - Bias							
1	31:2	Batch Buff	er Start Ad	dress								
		Format: GraphicsAddress[31:2]BatchBuffer										
		This field specifies Bits 31:2 of the starting address of the batch buffer.										
	1:0	Reserved				-						
		Format:	Format: MBZ									



	MI_CLFLUSH								
Source:	Source: RenderCS								
Length E	Length Bias: 2								
Flushes	out th	e page	given in the con	nmand o	out to sy	stem memory. This command is specific to the			
render	engine	e. This co	ommand is not p	brivilege	a.	Description			
Dword	BIT	C				Description			
0	51.29	Default	t Value:						
		Format				OpCode			
	28.23	MI Con	nmand Opcode			- F			
	20.25	Defaul	t Value:		27h Sto	re DW MI_CLFLUSH			
		Format	t:		OpCode				
	22	Use Glo	bal GTT	Per Proc	ess GTT	Enable hit is clear			
		Value	Name			Description			
		0h	Per Process Grapl	hics	This c	ommand will use the global PPGTT to translate the			
			Address		Addre	255.			
		1h	Global Graphics A	Address	This c Addre	This command will use the global GTT to translate the Address.			
	21:10	Reserve	ed						
		Format	t:			MBZ			
	9:0	DWord	Length						
		Defaul	t Value:	1h					
		Format	t:	=n Tota	l Length	- 2. Excludes DWord (0,1).			
1	31:12	Page B	ase Address	I					
		Format	t:	Graphic	sAddres	s[31:12]			
		4KB ali	gned Page Addres	ss which	software	requires hardware to flush to DRAM.			
	11:6	Startin	g Cacheline Offse	et					
		Format	t: U6 Zero base	d startin	g cachel	ne offset to the Page Base Address.			
	5:0	Reserve	ed						
		Format	t:			MBZ			
2n	31:0	DW Representing a Half Cache Line							
		Format: MBZ							
		The inf	ormation given	to hardv	vare is t	he DW itself, not the contents. Hardware uses the			
		DW co	unt of the comm	hand to $($	determi	ne the offset from the base to flush out. The			
		4096 b	vtes / 4 bvtes pe	er DW / 3	a WD 8	er HW = 128 DW.			
			,,, p.		Prog	ramming Notes			
		Always	even number of "	'DW Rep	resenting	g 1/2 cacheline" terms must be programmed.			



MI_CONDITIONAL_BATCH_BUFFER_END

Source: VideoCS

2

Length Bias:

The MI_CONDITIONAL_BATCH_BUFFER_END command is used to conditionally terminate the execution of commands stored in a batch buffer initiated using a MI_BATCH_BUFFER_START command. Termination of second level batch buffer due to this command will also terminate the parent/first level batch buffer.

level bat	er batch burier due to this command will also terminate the parent/first level batch buffer.								
				Programm	ning Notes				
This cor	his command is only valid with a 1st level batch buffer (bit 22 in MI_BATCH_BUFFER_START is set to 0).								
DWord	Bit	Description							
0	31:29	Command Ty	be		- 1				
		Default Value:			0h MI_COMMAND				
		Format:			OpCode				
	28:23	MI Command	Opcode	L					
		Default Value:		36h MI_CONDI	TIONAL_BATCH_B	UFFER_END			
		Format:		OpCode					
	22 Use Global GTT								
	Default Value:Oh DefaultValFormat:Boolean					esc			
Format: U1 FormatDesc						FormatDesc			
		If set, this com will be used to bit is clear.	mand will translate	use the global G the Compare Ad	iTT to translate th dress. This bit mu	e Compare Address. If ist be 1 if the Per Proce	clear, the PPGTT ss GTT Enable		
	21	Compare Sem	aphore						
		Default Value:	•		0h DefaultVaueD	esc			
		Format:			Boolean				
		If set, the value Address in mer execution of cu	et, the value from the Compare Data Dword is compared to the value from the Compare dress in memory. If the value at Compare Address is greater than the Compare Data Dword, ecution of current command buffer should continue. If clear, no comparison takes place.						
	20	Reserved							
	19:8	Reserved							
		Format: MBZ 7:0 DWord Length							
	7:0								
	Format: =n Total Length - 2								
		Value			Name				
		0h	Excludes	DWord (0,1) [De	fault]				
1	31.0	Compare Data	Dword		-				

Data dword to compare memory. The Data dword is supplied by software to control execution of the command buffer. If the compare is enabled and the data at Semaphore Address is greater



	MI_CLFLUSH										
		than this dword, the execution of the command buffer should continue.									
2	31:3	Compare Address Format: GraphicsAddress[31:3] Qword address to fetch compare Mask (DW0) and Data Dword(DW1) from memory. HW will do AND operation on Mask(DW0) with Data Dword(DW1) and then compare the result against Semaphore Data Dword									
	2:0	Reserved									
		Format:		MBZ							



MI_CONDITIONAL_BATCH_BUFFER_END

Source: RenderCS

2

Length Bias:

The MI_CONDITIONAL_BATCH_BUFFER_END command is used to conditionally terminate the execution of commands stored in a batch buffer initiated using a MI_BATCH_BUFFER_START command. Termination of second level batch buffer due to this command will also terminate the parent/first level batch buffer.

DWord	Bit		Description						
0	31:29	Command Type)						
		Default Value:			0h MI_COMMAND				
		Format:			OpCode				
	28:23	MI Command Opcode							
		Default Value:		36h MI_CONDITI	ONAL_BATCH_BUFFER_E	ND			
		Format:		OpCode					
	22	Use Global GTT	se Global GTT						
		Default Value:				0h			
		If set, this command will use the global GTT to translate the Compare Address . If clear, the PPGTT will be used to translate the Compare Address .							
	21	Compare Sema	phore						
		Default Value:				0h			
		If set, the value from the Compare Data Dword is compared to the value from the Compare							
		execution of current command buffer should continue. If clear, no comparison takes place.							
	20	Reserved							
	19:8	Reserved							
		Format:			MBZ				
	7:0	DWord Length	T				1		
		Format:	=n Tota	l Length - 2. Excl	udes DWord (0,1).				
		Value			Name				
		0h		[Default]					
1	31.0	Compare Data	Dword						
Ţ	51.0	Data dword to compare memory. The Data dword is supplied by software to com of the command buffer. If the compare is enabled and the data at Compare Addr than this dword, the execution of the command buffer should continue.							
2	2 31:3 Compare Address								
		Format:		GraphicsAddre	ess[31:3]				
		Format: Qword address	s to fetch	GraphicsAddre	ess[31:3] WWO) from memory.				
		Format: Qword address HW will compa	s to fetch are the D	GraphicsAddre n Data Dword(D Data Dword(DW	ess[31:3] DW0) from memory. 0) with Compare Data	Dword			



	MI_CONDITIONAL_BATCH_BU	JFFER_END
	Format:	MBZ



BlitterCS

2

MI_DISPLAY_FLIP

Source:

Length Bias:

The MI_DISPLAY_FLIP command is used to request a specific display plane to switch (flip) to display a new buffer. The buffer is specified with a starting address and pitch. The tiled attribute of the buffer start address is programmed as part of the packet.

The operation this command performs is also known as a "display flip request" operation - in that the flip operation itself will occur at some point in the future. This command specifies when the flip operation is to occur: either synchronously with vertical retrace to avoid tearing artifacts

Programming Notes

This command simply requests a display flip operation -- command execution then continues normally. There is no guarantee that the flip (even if asynchronous) will occur prior to subsequent commands being executed. (Note that completion of the MI_FLUSH_DW command does not guarantee that outstanding flip operations have completed). The MI_WAIT_FOR_EVENT command must be used to provide this synchronization to avoid back to back MI_DISPLAY_FLIP commands to the same display plane - by pausing command execution until a pending flip has actually completed. This synchronization can also be performed by use of the Display Flip Pending hardware status. See Display Flip Synchronization in the Device Programming Interface chapter of MI Functions.

After a display flip operation is requested, software is responsible for initiating any required synchronization with subsequent buffer clear or blitter operations. For multi-buffering (e.g., double buffering) operations, this will typically require updating SURFACE_STATE or the binding table to change the blitter (back) buffer. In addition, prior to any subsequent clear or blitter operations, software must typically ensure that the new blitter buffer is not actively being displayed. Again, the MI_WAIT_FOR_EVENT command or Display Flip Pending hardware status can be used to provide this synchronization. See Display Flip Synchronization in the Device Programming Interface chapter of MI Functions.

The display buffer command uses the X and Y offset for the tiled buffers from the Display Interface registers. Software is allowed to change the offset via the MMIO interface irrespective of the flip commands enqueued in the command stream. For tiled buffers, the display subsystem uses the X and Y offset in generation of the final request to memory. The offset is always updated on the next vblank for both Synchronous and Asynch Flips. It is not necessary to have a flip enqueued to update the X and Y offset

The display buffer command uses the linear DWord offset for the linear buffers from the Display Interface registers. Software is allowed to change the offset via the MMIO interface irrespective of the flip commands enqueued in the command stream. For linear buffers, the display subsystem uses the Dword offset in generation of the final request to memory.

- For synchronous flips the offset is updated on the next vblank. It is not necessary to have a sync flip enqueued to update the DWord offset.
- Linear memory does not support asynchronous flips.

The full packet must be contained within the same cache line.

There must be at least one valid command following this packet.

Events must be unmasked in the Display Engine Render Response Mask Register (DE RRMR 0x44050) prior to waiting for them with a MI_WAIT_FOR_EVENT command, or in the case of



MI_DISPLAY_FLIP

flips or scanlines, prior to starting the flip or loading the scanline. Unmasked events will wake command streamer as they occur, so for improved power savings it is recommended to only unmask events that are required. Programming the DE RRMR register can be done through MMIO or a LOAD_REGISTER_IMMEDIATE command.

DWord	Bit			Description				
0	31:29	Command Type						
		Default Value:		0h MI_COMMA	AND			
		Format: OpCode						
	28:23	MI Command Opcode						
		Default Value: 14h MI_DISPLAY			ELIP			
		Format:		OpCode				
	22	Async Flip Indicator						
		Format:		Enable	e			
		This bit should always be set if DW2 [1:0] == '01' (async flip). This field is required due to HW limitations. This bit is used by the blitter pipe while DW2 is used by the display hardware.						
	21:19	Display (Plane) Select This field selects which display plane is to perform the flip operation.						
		Value Name						
		0h	Oh Display Plane A					
		1h Display Plane B						
		2h Display Sprite A						
		3h Display Sprite B						
		4h	Display Pl	ane C				
		5h						
	18:17	Reserved			1			
		Format:			MBZ			
	16	Reserved			1			
		Format:			MBZ			
	15:13	Reserved			1			
		Format:			MBZ			
	12:8	Reserved						
		Format:			MBZ			
	7:0	DWord Length						
		Format:	=n Total L	ength - 2				
		For Synchronous Flips and As length of 3.	ynchronou	s Flips, this field m	nust be programmed to 1h for a total			
		Value		Na	me			



				Ν	/II_DISPLA	/_F	LIP			
		0h	E	xclude	s DWord (0,1) [Defa	ult]				
		1h	R	leserve	d					
1	31	Reserve	ed							
		Format	Format: MBZ							
	30:16	Reserve	eserved							
		Format	t:					MBZ		
	15:6	Reserve	ed							
	5:1	Reserve	ed					[
		Format	t:					MBZ		
	0	Tile Pa	Tile Parameter							
		Format	t:	F 1'			Enable	9 9		
		For Asy should or dire	maintain tl ct thru MM	he san IO.	this parameter car ne tile parameter a	nnot s pro	be cha ogram	med with the last synchronc	bus flip	
		Val	ue		Name			Description		
		0h	Linea	r [Def a	ault]	For	or Syncronous Flips Only			
		1h Tiled X								
					Program	min	a Note			
		Perform	ming a synch	ironou	s or asynchronous fli	n wil	l dron	any previous synchronous flin	that has	
		not yet	t completed.		o or asymetric croace in	P				
2	31:12	Display	/ Buffer Bas	e Addı	ress					
		Format	t:		GraphicsAddress[31	:12]				
		This fiel	d specifies E	Bits 31:	12 of the Graphics A	ddres	ss of th	e new display buffer.		
					Program	nmin	g Note	25		
		The Di	splay buffer	must r	eside completely in I	Main	Memo	ry.		
		This ac	dress is alwa	ays trai	nslated via the globa	l (rat	her tha	in per-process) GTT		
	11:3	Reserve	ed							
		Forma	-					MBZ		
	2	Reserv	ved							
	1:0 Flip Type This field specifies whether the flip operation should be performed asynchronously to vertic retrace.								rtical	
	Value Name Description									
		00b	Sync Flip [Default]	T a	he flip will occur dur woiding any tearing	ing t artifa	he vert cts.	ical blanking interval - thus		
		01b	Async Flip	T te	he flip will occur "as earing artifacts	soor	n as po	ssible" - and may exhibit		
		1b	Reserved							



			MI_FLUS	н				
Source:		RenderCS						
Length E	Bias:	1						
	Description							
on an i the rea data po • F b • I Usage comma flush is MI_FLU To use	 The MI_FLUSH command is used to perform an internal flush operation. The parser pauses on an internal flush until all drawing engines have completed any pending operations and the read caches are invalidated including the texture cache accessed via the Sampler or the data port. In addition, this command can also be used to: Flush any dirty data in the Render Cache to memory. This is done by default, however this can be inhibited. Invalidate the state and command cache. Usage Note: After this command is completed and followed by a Store DWord-type command, CPU access to graphics memory will be coherent (assuming the Render Cache flush is not inhibited). This command is specific to the render engine. Other engines use MI_FLUSH_DW. To use this command, bit 12 in the MI_MODE(0x209c) must be enabled. 							
If GFX_N address MI_FLU	MODE 0x4f1(SH con	(0x229C) bit 13, this comma 00. hmand is no longer validate	and will cause a config ed or supported. Use a	y write to MMIO register space with the at your own risk.				
DWord	Bit		Desc	ription				
0	31:29	Command Type						
		Default Value:	Oh M	I_COMMAND				
		Format:	Ορία	bde				
	28:23	MI Command Opcode						
				040 MI_FLOSH				
		Format.		Opcode				
	22:7	Reserved		MP7				
	6			INDZ				
	6 F	Reserved						
	5	Format:	ISADIe	Disable				
		At the completion of the considered as invalid. I.e	e flush, the indirect s e., the indirect point	state pointers in the hardware will be ers will not be restored for the context.				
	4	Generic Media State Clea	ar					
		Format:		Disable				
If set, all generic media state context information will not be included with the ne								

context save, assuming no new state is initiated after the flush. If clear, the generic media state context save state will not be affected. An MI_FLUSH with this bit set should be issued once all the Media Objects that will be processed by a given persistent root



		MI_DI	SPLAY	_FLIP		
	(T switching from a generic media ET_CONTEXT, once state is any context each time that context is issued in that context.					
3	Reserved					
2	Render Ca					
	Format:			Boolean		
	If set, the F	ender Cache is not flushe	hed as part of the processing of this command.			
	Value	Name		Description		
	0h	Flush	Flush the	Flush the Render Cache.		
	1h	Don't Flush	Do not flush the Render Cache.			
1	State/Instruction Cache Invalidate					
	Format:			Boolean		
	If set, Inval	idates the State and Instr	uction Cach	e.		
	Value	Name			Description	
	0h	Don't Invalidate	Leave State	/Instruction	n Cache unaffected.	
	1h	Invalidate	Invalidate S	state/Instru	ction Cache.	
0	Reserved					
	Format:				MBZ	



MI_FLUSH_DW

Source: BlitterCS

Length Bias:

The MI_FLUSH_DW command is used to perform an internal "flush" operation. The parser pauses on an internal flush until all drawing engines have completed any pending operations. In addition, this command can also be used to: Flush any dirty data to memory. Invalidate the TLB cache inside the hardware

Usage note: After this command is completed with a Store DWord enabled, CPU access to graphics memory will be coherent (assuming the Render Cache flush is not inhibited).

DWord	Bit	Description						
0	31:29	Command Type						
		Default Value:	0h MI_COMMAND					
	28:23	MI Command Opcode						
		Default Value:	26h MI_FLUSH_DW					
	22	Reserved						
		Format:	U	J1				
	21	Store Data Index						
		Format:	U	J1				
		This field is valid only if the post-sync	operation is not 0. If thi	is bit is set, the store data				
		address is actually an index into the h	ardware status page.					
	20.10							
	20:19	Reserved	NAD7					
	10							
	18	TLB Invalidate		11				
		Format:	U	JI				
		Des	cription					
		If ENABLED, all TLBs belonging to Blitter	Engine will be invalidated	once the flush				
		value of 1h or 3h.	ia when the rost syne op					
		If GFX_MODE (0x229c) bit 13, this comm	and will cause a config wri	ite to MMIO				
		register space with the address 0x4f100.						
	17	Synchronize GFDT surface		1				
		Format:	U	J1				
		If enabled, at the end of the current flush the last level cache is cleared of all the cachelines which have been marked with the special GFDT flags. Store DW must be enabled						
	16	Reserved						
		Format:	MBZ					
	15:14	Post-Sync Operation BitFieldDesc						



			IVII_FLUSH_DVV				
	Value	Name	Description				
	0h	No Write	No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc.				
	1h	Write Immedia Data QWord	Write the QWord containing Immediate Data Low, High DWs to the Destination Address				
	2h	Reserved	Reserved				
	3h		Write the TIMESTAMP register to the Destination Address with a granularity of 80ns. The upper 28 bits of the TIMESTAMP register are tied to '0'.				
			Programming Notes				
	If exects space.	uted in a non- . If in a secure	-secure batch buffer, the address given is in a PPGTT address ring or batch, the address given is in GGTT space.	S			
13:10	Reserve	ed					
	Format	t:	MBZ				
9	Reserve	ed					
	Format: MBZ						
8	Notify Enable						
	Forma	t:	U1				
	If ENABLED, a Sync Completion Interrupt will be generated (if enabled by the MI Interrup Control registers) once the sync operation is complete. See Interrupt Control Registers in Memory Interface Registers for details.						
7:6	Reserve	ed					
7:6	Reserve Format	ed t:	MBZ				
7:6	Reserve Format	ed t: Length	MBZ				
7:6	Reserve Format	ed t: I Length t:	MBZ =n Total Length - 2				
7:6	Reserve Format DWord Format	ed t: I Length t:	MBZ =n Total Length - 2 Name				
7:6	Reserve Format DWord Format Value 2h	ed t: I Length t: Excludes DW	MBZ = n Total Length - 2 Name /ord (0,1) = 1 for DWord, 2 for QWord [Default]				
7:6 5:0 31:3	Reserv Forma DWord Forma Value 2h Addres	ed t: I Length t: Excludes DW	MBZ = n Total Length - 2 Name Yord (0,1) = 1 for DWord, 2 for QWord [Default]				
7:6 5:0 31:3	Reserve Formation DWord Formation Value 2h Address Formation	ed t: I Length t: Excludes DW s t: d specifies Bits	MBZ =n Total Length - 2 Name /ord (0,1) = 1 for DWord, 2 for QWord [Default] GraphicsAddress[31:3]U28 31:3 of the Address where the DWord or OWord will be stored. No				
7:6 5:0 31:3	Reserv Format DWord Format Value 2h Addres Format This fiel that the	ed t: I Length t: Excludes DW s t: d specifies Bits address can or	Image: MBZ Image: Ima	ote			
7:6 5:0 31:3 2	Reserv Forma DWord Forma Value 2h Addres Forma This fiel that the Destina Defines	ed t: I Length t: Excludes DW/ s t: d specifies Bits address can or ation Address T address space	Image: MBZ Image: Ima	ote			
7:6 5:0 31:3 2	Reserv Forma DWord Forma Value 2h Addres Forma This fiel that the Defines Valu	ed t: I Length t: Excludes DW s t: d specifies Bits address can or stion Address T address space Name	Image: MBZ Image: Ima	ote			



	MI_FLUSH_DW							
		1h	Ih GGTT Use GGTT address space for DW write					
			Programming Notes					
		Ignored if	"No write"	is the selected in Operation.				
	1:0	Reserved						
		Format:			MBZ			
23	31:0	Immediate	e Data					
		This field specifies the DWord value to be written to the targeted location. DW2 is the lower DW if QW is desired. Only valid when 15:14 in header is set to 1h						
		To avoid h address is	nitting a kno '1'	wn hardware bug, drivers cannot se	end a QW write when bit 5 of the			



VideoCS

2

MI_FLUSH_DW

Source:

Length Bias:

The MI_FLUSH_DW command is used to perform an internal "flush" operation. The parser pauses on an internal flush until all drawing engines have completed any pending operations. In addition, this command can also be used to:Flush any dirty data to memory. Invalidate the TLB cache inside the hardware Usage note: After this command is completed with a Store DWord enabled, CPU access to graphics memory will be coherent (assuming the Render Cache flush is not inhibited).

DWord	Bit		Description				
0	31:29	Comma	and Type				
		Defaul	t Value:		0h MI_COMMAND		
	28:23	MI Con	nmand Opcode				
		Defaul	Default Value: 26h MI_FLUSH_DW				
	22	Reserv	ed				
	21	Store D	Data Index				
		Forma	t:			U1	
		This fiel is actua	This field is valid only if the post-sync operation is not 0. If this bit is set, the store data address is actually an index into the hardware status page.				
	20:19	Reserv	ed				
		Forma	t:		MBZ		
	18	TLB Inv	validate				
		Forma	t:			U1	
		Description					
		If ENABLED, all TLBs belonging to Video Engine will be invalidated once the flush operation is complete. This bit is only valid when the Post-Sync Operation field is a value of 1h or 3h.					
		If GFX_MODE(0x229c) bit 13, this command will cause a config write to MMIO register space with the address 0x4f100.					
	17	Synchr	onize GFDT surf	ace			
		Forma	t:			U1	
		If enabled, at the end of the current flush the last level cache is cleared of all the cachelines which have been marked with the special GFDT flags. Store DW must be enabled					
	16	Reserv	ed				
		Forma	t:		MBZ		
	15:14 Post-Sync Operation BitFieldDesc						
Value Name Description							
		0h	No Write	No write occurs a used to impleme	as a result of this instructed a "trap" operation, e	ction. This can be tc.	



				N	II_FLUSH_DW					
		1hWrite Immediate DataHW implicitly detects the Data size to be Qword or Dword to be written to memory based on the command dword length programmed . When Dword Length indicates Qword, Writes the QWord containing Immediate Data Low, High DWs to the Destination Address . When Dword Length indicates Dword, Writes the DWord containing Immediate Data Low to the Destination Address								
		2h	Reserved	Res	Reserved					
		3h		Wri wit The	Write the TIMESTAMP register to the Destination Address with a granularity of 80ns. The upper 28 bits of the TIMESTAMP register are tied to '0'.					
	13:10	Reserve Format	ed ::		MB7					
	9	Reserved								
		Format: MBZ								
	8	Notify	Enable					I		
Format:							U1			
		If ENABLED, a Sync Completion Interrupt will be generated (if enabled by the MI Interru Control registers) once the sync operation is complete. See Interrupt Control Registers i Memory Interface Registers for details.					bled by the MI Interru ipt Control Registers i	pt n		
	/	VIGEO H	² ipeline Cache	Inval	lidate		111			
		Enable	 the invalidatior	n of th	e video cache at the end of this	flush	01			
	6	Reserve	ed							
	5:0	DWord	Length							
		Format	:		=n Total Length - 2					
		Value	Evoludes DM	lard ((Name					
_		20	Excludes DW		(J,I) = I for Dword, 2 for Qword					
1	31:3	Addres	S 	Cro	nhias Addross [21,2]1120					
		This fiel	 d snecifies Bits	31.3	of the Address where the DWork	d or O	Word will be stored. N	lote		
	that the address can only be QWord aligned, irrespective of data size.					size.				
	2	Destina	address space	Type	estination Address					
		Valu	ie Name		Desc	riptio	n			
		0h	PPGTT	l	Jse PPGTT address space for DW	V write				
		1h	GGTT	ι	Jse GGTT address space for DW	write				



	MI_FLUSH_DW								
		Programming Notes Ignored if "No write" is the selected in Operation.							
	1:0	Reserved							
		Format:	MBZ						
23	31:0	Immediate Data							
		This field specifies the DWord value to be written to the targeted location. DW2 is the lower DW if QW is desired. Only valid when 15:14 in header is set to 1h							
		To avoid hitting a known hardware bug, drivers cannot s address is '1'	end a QW write when bit 5 of the						



MI_LOAD_REGISTER_IMM

Source:		BlitterCS				
Length Bias: 2						
The MI_LOAD_REGISTER_IMM command requests a write of up to a DWord constant supplied in the command to the specified Register Offset (i.e., offset into Memory-Mapped Register Range). The register is loaded before the next command is executed.						
DWord	Bit	Description				
0	31:29	Command Type				
		Default Value:		0h MI_COMMAND		
	28:23	MI Command Opcode				
		Default Value:			22h MI_	
	22:12	Reserved				
		Format:			MBZ	
	11:8	Byte Write Disables				
		Format: Enable[4] Bit 8 corresponds to Data DWord [7:0]				
		Range: Must specify a valid register write operation				
		If [11:8] is '1111b', then the register write will not occur. If [11:8] is '0000b', then the register DW will be updated. Any other value, the behavior will be specifically specified by the register or the behavior is undefined.				
	7.0	DWord Length				
	7.0	Default Value:	It Value:		1h Excludes DWord (0.1)	
		Format:		=n Total Length - 2		
1	31:23	Reserved				
-		Format:		ſ	MBZ	
	22:2	Register Offset				
		Format:	Format: U21			
		Format:	Mmio	MmioAddress[22:2]		
		This field specifies bits [22:2] of the offset into the Memory Mapped Register Range (i.e., this field specifies a DWord offset).				
	1:0	Reserved				
		Format:			MBZ	
2	31:0	Data DWord				
		Mask: Bytes Write Disables		Write Disables		
		Format:	nat: U32			
		This field specifies the DWord value to be written to the targeted location.				


MI_LOAD_REGISTER_IMM

Source: RenderCS

2

Length Bias:

The MI_LOAD_REGISTER_IMM command requests a write of up to a DWord constant supplied in the command to the specified Register Offset (i.e., offset into Memory-Mapped Register Range).

Programming Notes

A stalling flush must be sent down pipeline before issuing this command.

To ensure this command gets executed before upcoming commands in the ring, either a stalling pipeControl should be sent after this command, or MMIO 0x20C0 bit 7 should be set to 1.

The following addresses should NOT be used for LRIs:

- 1. 0x8800 0x88FF
- 2. >= 0xC0000

Limited LRI cycles to the Display Engine 0x40000-0xBFFF) are allowed, but must be spaced to allow only one pending at a time. This can be done by issuing an SRM to the same address immediately after each LRI.

MI_LOAD_REGISTER_IMM command to program Scanline Register followed by Wait For Event command with Scanline Wait, should always be programmed in the same cacheline together without any commands (including pipe control) in between and also should be submitted in the same ring dispatch.

DWord	Bit	Description					
0	31:29	Command Ty	/pe				
		Default Value	e:			0h MI_COMMA	AND
		Format:				OpCode	
	28:23	MI Comman	d Opcode				
		Default Value	9:		22h MI_L	OAD_REGISTER	_IMM
		Format:			OpCode		
	22:13	Reserved					
		Format:					MBZ
	12	Reserved					
	11:8	Byte Write D	isables				
		Format:	Enable[4]] Bit 8 d	correspon	ds to Data DWc	ord [7:0]
		Range: Must	specify a v	alid re	gister writ	te operation	
		If [11:8] is '11	If [11:8] is '1111b', then this command will behave as a NOOP.				
		Otherwise, the value is forwarded to the destination register.					
	7:0	DWord Leng)Word Length				
		Default Value	e:	1h Ex	cludes DV	/ord (0,1)	
		Format:		=n To	otal Lengtl	n - 2. Excludes D	DWord (0,1).



	MI_LOAD_REGISTER_IMM									
1	31:23	Reserved								
		Format: MBZ								
	22:2	Register Offset								
		Format:	MmioAddress[22:2]							
		This field specifies bits [22:2] of the offset into the Memory Mapped Register Range (i.e., this field specifies a DWord offset).								
	1:0	Reserved								
		Format:	MBZ							
2	31:0	Data DWord								
		Mask:	Bytes Write Disables							
		Format:	U32							
		This field specifies the DW	ord value to be written to the targeted location.							



MI_LOAD_REGISTER_IMM

Source:

VideoCS Length Bias: 2

The MI_LOAD_REGISTER_IMM command requests a write of up to a DWord constant supplied in the command to the specified Register Offset (i.e., offset into Memory-Mapped Register Range). The register is loaded before the next command is executed.

DWord	Bit			Description		
0	31:29	Command Type				
		Default Value:		0h MI_COMMAND		
		Format:		OpCode		
	28:23	MI Command Opcode				
		Default Value:	22h MI_L	OAD_REGISTER_IMM		
		Format:	OpCode			
	22:12	Reserved				
		Format:		MBZ		
	11:8	Byte Write Disables				
		Format: Enable[4] (bit 8	correspon	ds to Data DWord [7:0]).		
		Range: Must specify a valid re	egister writ	e operation		
		If [11:8] is '1111b', then the re	egister write	e will not occur.		
		Any other value, the behavio	egister DW or will be so	/ will be updated.		
		undefined.				
	7:0	DWord Length				
		Default Value:	0h Ex	cludes DWord (0,1)		
		Format:	=n Te	otal Length - 2		
1	31:23	Reserved				
		Format:		MBZ		
	22:2	Register Offset				
		Format:	MmioAddre	ess[22:2]		
		This field specifies bits [22:2] of	of the offse	t into the Memory Mapped Register Range (i.e., this		
		field specifies a DWord offset). Mapped				
	1:0	Reserved				
		Format:		MBZ		
2	31:0	Data DWord				
		Format:	U32 Forr	matDesc		
		This field specifies the DWord	value to be	e written to the targeted location.		



MI_LOAD_REGISTER_MEM

RenderCS, BlitterCS, VideoCS Source: 2

Length Bias:

The MI_LOAD_REGISTER_MEM command requests from a memory location and stores that DWord to a register.

Programming Notes

The command temporarily halts commands that will cause cycles down the 3D pipeline.

The following addresses should NOT be used for LRIs:

- 0x8800 0x88FF
- >= 0xC0000

1

Limited LRI cycles to the Display Engine 0x40000-0xBFFFF) are allowed, but must be spaced to allow only one pending at a time. This can be done by issuing an SRM to the same address immediately after each LRI.

Any updates to the memory location exercised by this command must be ensured to be coherent in memory prior to programming of this command. This must be achieved by programming "16" dummy MI_STORE_DATA_IMM (write to scratch space) commands prior to programming of this command. Example:

MI_STORE_REGISTE_MEM (0x2288, 0x2CF0_0000)

.

MI STORE DATA IMM (16 times) (Dummy data, Scratch Address) MI_LOAD_REGISTER_MEM(0x2288, 0x2CF0_0000)

DWord	Bit				Description		
0	31:29	Comma	and Type				
		Defaul	t Value:		0h MI_COMMAND		
		Format	t:		OpCode		
	28:23	MI Con	nmand Opcode				
		Defaul	t Value:	29h MI_L	OAD_REGISTER_MEM		
		Format	t:	OpCode			
	22	Use Glo This bit	bbal GTT must be 1 if the Per Pro	cess GTT	Enable bit is clear.		
		Value	Name		Description		
		0h	Per Process Graphics Address	This Addr	This command will use the per process GTT to translate the Address.		
		1h	Global Graphics Addres	s This Addr	This command will use the global GTT to translate the Address.		
21 Async Mode Enable If this bit is set then the command stream will not wait for comple executing the next command. Please refer to the LOAD_INDIRECT usage of this bit.				n will not wait for completion of this command before er to the LOAD_INDIRECT and Predicate registers for			
	20:8	Reserve	ed				
		Format	t:		MBZ		

٦



		MI_LC	DAD_R	EGISTER_M	IEM			
	7:0	DWord Length	DWord Length					
		Default Value:	1h Excludes	5 DWord (0,1)				
		Format:	=n Total Le	ngth - 2. Excludes D	Word (0,1).			
1	31:23	Reserved						
		Format:			MBZ			
	22:2	Register Address						
		Format:	MMIOA	ddress[22:2]				
		This field specifies Bits 22 address must be DWord	2:2 of the Re -aligned, Bits	gister offset the DWo 5 1:0 of that address I	ord will be written to. As the register MBZ.			
	1:0	Reserved						
		Format:			MBZ			
2	31:2	Memory Address						
		Format:	GraphicsA	ddress[31:2]				
		This field specifies the ac DWord above will read fi Range = GraphicsVirtual	his field specifies the address of the memory location where the register value specified in the Word above will read from. The address specifies the DWord location of the data. ange = GraphicsVirtualAddress[31:2] for a DWord register					
	1:0	Reserved						
		Format:			MBZ			

Command Reference - Instructions



				MI_NC	OOP			
Source:		BlitterC	S					
Length Bi	as:	1						
The MI_NOOP command basically performs a "no operation" in the command stream and is typically used to pad the command stream (e.g., in order to pad out a batch buffer to a QWord boundary). However, there is on minor (optional) function this command can perform - a 22-bit value can be loaded into the MI NOPID register. This provides a general-purpose command stream tagging ("breadcrumb") mechanism (e.g., to provide sequencing information for a subsequent breakpoint interrupt).						command stream and is typically used to o a QWord boundary). However, there is one e can be loaded into the MI NOPID register. crumb") mechanism (e.g., to provide		
DWord	Bit				Descript	tion		
0	31:29	Command Default Va	Type lue:		0h MI_C	COMMAND		
	28:23	MI Comma	nd Opcode					
		Default Va	lue:		(0h MI_NOOP		
	22	Identificati	on Number P	legister Write F	Enable			
		Format:				Enable		
		This field er register. If c operation"	ables the valu lisabled, that r function.	e in the Identifi egister is unmo	cation Nu dified - n	umber field to be written into the MI NOPID naking this command an effective "no		
		Value	Name		De	escription		
		0h	Disable	Do not write the NOP_ID register.				
		1h	Enable	Write the NOP_ID register.				
	21:0	Identificati	on Number					
		Format:				U22		
		This field co	ontains a 22-bi	t number which	i can be v	written to the MI NOPID register.		



MI_NOOP

Source: RenderCS

1

Length Bias:

The MI_NOOP command basically performs a "no operation" in the command stream and is typically used to pad the command stream (e.g., in order to pad out a batch buffer to a QWord boundary). However, there is one minor (optional) function this command can perform - a 22-bit value can be loaded into the MI NOPID register. This provides a general-purpose command stream tagging ("breadcrumb") mechanism (e.g., to provide sequencing information for a subsequent breakpoint interrupt).

Performance

The MI_NOOP process time is reduced to 1 clock. An example use of the improved NOOP throughput is for some multi-pass media applications where some unwanted media object commands are replaced by MI_NOOP commands without repacking the commands in a batch buffer.

DWord	Bit				Descrip	otion	
0	31:29	Command Ty	уре				
		Default Value	e:		0h MI_0	COMMAN	D
	28:23	MI Comman	MI Command Opcode				
		Default Value	e:			0h MI_NC	OOP
	22	Identification Number Register Write Enable					
		Format:				Enable	
		This field ena	ables the value in	າ the Identif	ication N	Number fie	eld to be written into the MI NOPID
		register. If dis	abled, that regis	ter is unmo	dified, m	naking this	command an effective "no
		operation" tu	nction.				
		Value	Name			De	scription
		0h	Disable	Do not wr	ite the N	IOP_ID reg	ister.
		1h	Enable	Write the	NOP_ID	register.	
	21:0	Identification Number					
		Format: U22					U22
		This field cor	ntains a 22-bit nu	umber which	h can be	written to	the MI NOPID register.

Command Reference - Instructions



			MI_NC	DOF				
Source:		VideoCS						
Length Bi	as:	1						
The MI_NOOP command basically performs a "no operation" in the command stream and is typically used to pad the command stream (e.g., in order to pad out a batch buffer to a QWord boundary). However, there is one minor (optional) function this command can perform - a 22-bit value can be loaded into the MI NOPID register. This provides a general-purpose command stream tagging ("breadcrumb") mechanism (e.g., to provide sequencing information for a subsequent breakpoint interrupt).								
DWord	Bit			Desci	ript	tion		
0	31:29	Command Type						
		Default Value:		0h M	I_C	COMMAND		
		Format:		ОрСо	ode	e		
	28:23	MI Command Opco	ode					
		Default Value:			00	0h MI_NOOP		
		Format:			Op	pCode		
	22	Identification Num	ber Register Write I	Enable)			
		Format:				Enable		
		This field enables the value in the Identification Number field to be written into the MI NOPID register. If disabled, that register is unmodified - making this command an effective "no						
		Value	Value Name					
		1	Write the NOP ID register.					
	21.0	Identification Num	ber	<u> </u>			_	
	22.0	Format:				U22	٦	
		This field contains a	22-bit number which	can b	e v	written to the MI NOPID register.		



Source: RenderCS Length Bias: 1 DWord Bit Description 0 31:29 Command Type Default Value: Oh ML_COMMAND Format: OpCode 28:23 MI Command Opcode Default Value: Och ML_PREDICATE Format: OpCode 22:8 Reserved Format: MBZ 7.6 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Description Oh LOADOP_LOAD The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the current Predicate state bit. Yalue Name Description 0h COMBINEOP_SET The combine operation output the compare result and the current Predicate state bit. Yalue Name Description 0h COMBINEOP_OR The combine operation outputs the AND of the compare result and the current Predicate state bit. <t< th=""><th></th><th colspan="6">MI_PREDICATE</th></t<>		MI_PREDICATE								
Length Bias: 1 DWord Bit Description 0 31:29 Command Type Default Value: Oh ML_COMMAND Format: OpCode 28:23 MI Command Opcode Default Value: Och ML_PREDICATE Format: OpCode 22:38 Reserved Format: OpCode MBZ 7:6 Load Operation This field controls if/how the Predicate state bit is modified. MBZ 7:6 Load Operation This field controls if/how the Predicate state bit is loaded with the combine operation result. 3:h LOADOP_LOAD The Predicate state bit is loaded with the combine operation result. 3:h LOADOP_LOAD The Predicate state bit is loaded with the inverted combine operation result. 3:h LOADOP_LOAD The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description Oh 0:h COMBINEOP_SET The combine operation outputs the AND of the compare result and the current Predicate state bit. Value Name Description Oh OMBINEOP_COR 1:h </th <th>Source:</th> <th></th> <th>R</th> <th>enderCS</th> <th></th> <th></th> <th></th>	Source:		R	enderCS						
DWord Bit Description 0 31:29 Command Type Default Value: 0h ML_COMMAND Format: 0pCode 0pCode 28:23 MI Command Opcode Default Value: 0Ch ML_PREDICATE Format: 0pCode 0pCode 22:8 Reserved Format: 0pCode 7:6 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Description 0h LOADOP_KEEP The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOAD The Predicate state bit is loaded with the inverted combine operation result. 3h LOADOP_LOAD The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Image: Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation outputs the AND of the compare result and the current Predicate state bit. Value Name Description 0h COMBINEOP_AND The combine operation outputs the AND of th	Length B	Bias:	1							
DWord Bit Description 0 31:29 Command Type Default Value: Oh MI_COMMAND Format: OpCode OpCode 28:23 MI Command Opcode Default Value: Och MI_PREDICATE Format: OpCode 22:8 Reserved Format: OpCode 22:8 Reserved Format: MBZ 7:6 Load Operation This field controls if/how the Predicate state bit is modified. MBZ 7:6 Load Oper_LOAD The Predicate state bit is unmodified. 1h Reserved Description 2h LOADOP_LCAD The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description 0h COMBINEOP_SET The combine operation output the compare result and the current Predicate state bit. Value Name Descript										
0 31:29 Command Type Default Value: 0h MI_COMMAND Format: OpCode 28:23 MI Command Opcode Default Value: 0Ch MI_PREDICATE Format: OpCode 22:8 Reserved Format: OpCode 7:6 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Description 0h LOADOP_KEEP The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOAD The Predicate state bit is loaded with the inverted combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the current predicate state bit. 5 Reserved MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description 0h COMBINEOP_SET The combine operation outputs the AND of the compare result and the current Predicate state bit. 1h COMBINEOP_	DWord	Bit	Description							
Default Value: Oh MI_COMMAND Format: OpCode 28:23 MI Command Opcode Default Value: OCh MI_PREDICATE Format: OpCode 22:8 Reserved Format: OpCode 7:5 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Oh LOADOP_KEEP 7:6 LOADOP_LOAD 2h LOADOP_LOAD 3h LOADOP_LOADINV The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOADINV The Predicate state bit. Ioaded with the inverted combine operation result. 3h LOADOP_LOADINV The Predicate state bit. MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description 0h COMBINEOP_SET The combine operation outputs the AND of the compare result and the current Predicate state bit. 1h COMBINEOP_AR	0	31:29	Comma	and Type						
Format: OpCode 28:23 MI Command Opcode Default Value: OCh MI_PREDICATE Format: OpCode 22:8 Reserved Format: MBZ 7:6 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Obscription 0h LOADOP_KEEP The Predicate state bit is unmodified. 1h Reserved 2h LOADOP_LOAD 3h LOADOP_LOAD 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description 0h COMBINEOP_SET The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_AND The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The			Defaul	t Value:		0h MI_COMMA	ND			
28:23 MI Command Opcode Default Value: OCh MI_PREDICATE Format: OpCode 22:8 Reserved Format: MBZ 7:6 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Obscription 0h LOADOP_KEEP 1h Reserved 2h LOADOP_LOAD 3h LOADOP_LOAD 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: MBZ MBZ 4/3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Value Value Value Value Value Value Value Value Value <td colsp<="" td=""><th></th><td></td><td>Forma</td><td>t:</td><td></td><td>OpCode</td><td></td></td>	<th></th> <td></td> <td>Forma</td> <td>t:</td> <td></td> <td>OpCode</td> <td></td>			Forma	t:		OpCode			
Default Value: Och ML_PREDICATE Format: OpCode 22:8 Reserved Format: MBZ 7:6 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Operation Oh LOADOP_KEEP The Predicate state bit is unmodified. 1h Reserved 2h LOADOP_LOAD ADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Oh COMBINEOP_SET The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR		28:23	MI Command Opcode							
Format: OpCode 22:8 Reserved Format: MBZ 7:6 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Description 0h LOADOP_KEEP The Predicate state bit is unmodified. 1h Reserved Image: Control if /how the Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOAD The Predicate state bit is loaded with the inverted combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 3h Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. 4:3 Combine Operation The combine operation outputs the AND of the compare result and the current Predicate state bit. 1h COMBINEOP_SET The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_NOR The combine operation outputs the OR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR			Defaul	t Value:		0Ch MI_PREDICA	ATE			
22:8 Reserved Format: MBZ 7:6 Load Operation This field controls if/how the Predicate state bit is modified. Value Value Name Description 0h LOADOP_KEEP The Predicate state bit is unmodified. 1h Reserved Image: Control operation 2h LOADOP_LOAD The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved MBZ 4:3 Combine Operation MBZ 1h Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description 0h COMBINEOP_SET The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_AND The combine operation outputs the OR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR of the compare res			Forma	t:		OpCode				
Format: MBZ 7:6 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Description 0h LOADOP_KEEP The Predicate state bit is unmodified. 1h Reserved Image: Combine operation result. 2h LOADOP_LOAD The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description 0h COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs		22:8	Reserv	ed						
7:6 Load Operation This field controls if/how the Predicate state bit is modified. Value Name Oh LOADOP_KEEP The Predicate state bit is unmodified. 1h Reserved 2h LOADOP_LOAD 3h LOADOP_LOADINV The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name 0h COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the Compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR			Forma	t:			MBZ			
Value Name Description 0h LOADOP_KEEP The Predicate state bit is unmodified. 1h Reserved		7:6	Load O This fie	peration eld controls if/how th	ne Predicate	e state bit is modifie	d.			
0h LOADOP_KEEP The Predicate state bit is unmodified. 1h Reserved			Value	Name		D	escription			
1h Reserved 2h LOADOP_LOAD The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: MBZ 4:3 Combine Operation MBZ 4:3 Combine Operation MBZ 4:3 Combine Operation MBZ 4:4 Name Description 0h COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit.			0h	LOADOP_KEEP	The Predic	The Predicate state bit is unmodified.				
2h LOADOP_LOAD The Predicate state bit is loaded with the combine operation result. 3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name 0h COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. <th></th> <td></td> <td>1h</td> <td>Reserved</td> <td></td> <td colspan="4"></td>			1h	Reserved						
3h LOADOP_LOADINV The Predicate state bit is loaded with the inverted combine operation result. 5 Reserved Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Oh COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2 Reserved Format: MBZ			2h	LOADOP_LOAD	ed with the combine operation result.					
5 Reserved Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description 0h COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2 Reserved Format: MBZ			3h	LOADOP_LOADINV	/ The Predicate state bit is loaded with the inverted combine operation result.					
Format: MBZ 4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description 0h COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2 Reserved Format: MBZ		5	Reserv	ed						
4:3 Combine Operation This field controls if/how the result of the compare operation is combined with the current Predicate state bit. Value Name Description 0h COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2 Reserved Format: MBZ			Forma	t:			MBZ			
Value Name Description 0h COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2 Reserved The combine operation outputs the XOR of the compare result and the current Predicate state bit.		4:3	Combi This fie Predica	ne Operation Id controls if/how th te state bit.	ne result of	the compare operat	ion is combined with the current			
0h COMBINEOP_SET The combine operation output the compare result unmodified. 1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2 Reserved The combine operation outputs the XOR of the compare result and the current Predicate state bit.			Value	Name		Description				
1h COMBINEOP_AND The combine operation outputs the AND of the compare result and the current Predicate state bit. 2h COMBINEOP_OR The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2 Reserved Format: MBZ			0h	COMBINEOP_SET	The combi	ne operation output	t the compare result unmodified.			
2h COMBINEOP_OR The combine operation outputs the OR of the compare result and the current Predicate state bit. 3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2 Reserved Format: MBZ			1h	COMBINEOP_AND	The combine the current	ne operation output t Predicate state bit.	ts the AND of the compare result and			
3h COMBINEOP_XOR The combine operation outputs the XOR of the compare result and the current Predicate state bit. 2 Reserved Format: MBZ			2h	COMBINEOP_OR	The combine the current	ne operation output t Predicate state bit.	ts the OR of the compare result and			
2 Reserved Format: MBZ		3h COMBINEOP_XOR The combine operation outputs the XOR of the compare results the current Predicate state bit.					ts the XOR of the compare result and			
Format: MBZ		2	Reserve	ed						
			Forma	t:			MBZ			
1:0 Compare Operation This field controls how Data DWord 0 and Data DWord 1 fields are used to generate a compare operation result and possibly modify the PredicateData register		1:0	Compa This fie	re Operation eld controls how Data	a DWord 0	and Data DWord 1	fields are used to generate a compare			
Value Name Description			Value	Name	y mouny ti		Description			



MI_PREDICATE								
0h	COMPAREOP_TRUE	The compare operation outputs TRUE. The PredicateData register is unmodified.						
1h	COMPAREOP_FALSE	The compare operation outputs FALSE. The PredicateData register is unmodified.						
2h	COMPAREOP_SRCS_EQUAL	(MItemp0 - MItemp1) is computed and loaded into the PredicateData register. The compare operation outputs (MItemp0 == MItemp1).						
3h	COMPAREOP_DELTAS_EQUAL	(MItemp0 - MItemp1) is computed and compared to the PredicateData register. If the values are equal, the compare operation outputs TRUE, otherwise it outputs FALSE. The PredicateData register is unmodified.						



MI_REPORT_HEAD

Source: BlitterCS

1

Length Bias:

The MI_REPORT_HEAD command causes the Head Pointer value of the active ring buffer to be written to a cacheable (snooped) system memory location.

 Programming Notes

 This command must not be executed from a Batch Buffer (Refer to the description of the HWS_PGA register).

 DWord
 Bit
 Description

 0
 31:29
 Command Type

 Default Value:
 0h MI_COMMAND

	Default Value:	Oh MI_COMI	MAND
28:23	MI Command Opcode		
	Default Value:	07h MI_REPORT	_HEAD
22:0	Reserved		
	Format:		MB7

1



MI_REPORT_HEAD

Source: RenderCS

Length Bias:

The MI_REPORT_HEAD command causes the Head Pointer value of the active ring buffer to be written to a cacheable (snooped) system memory location. The location written is relative to the address programmed in the Hardware Status Page Address Register.

Programming Notes

This command must not be executed from a Batch Buffer. (Refer to the description of the HWS_PGA register.)

				-	
DWord	Bit	Description			
0	31:29	Command Type			
		Default Value: 0h MI_COMMAND			
		Format: OpCode			
	28:23	MI Command Opcode			
		Default Value:	07h MI_REPORT	_HEAD	
		Format: OpCode			
	22:0	Reserved			
		Format:		MBZ	



MI_REPORT_HEAD

Source: VideoCS

1

Length Bias:

The MI_REPORT_HEAD command causes the Head Pointer value of the ring buffer to be written to a cacheable (snooped) system memory location.

	Programming Notes					
This command m	This command must not be executed from a Batch Buffer (Refer to the description of the HWS_PGA register).					
DWord	Bit	Description				
0	31:29	Command Type				
		Default Value: 0h MI_CO		MAND		
		Format:	OpCode	OpCode		
	28:23	MI Command Opcode				
		Default Value:	07h MI_REPOR	T_HEAD		
		Format:	OpCode			
	22:0	Reserved	Reserved			
		Format:		MBZ		



MI	SEMA	PHORE	MBOX
_		_	

Source:

Length Bias:

BlitterCS

2

Description

MI_SEMAPHORE_MBOX command provides capability in Blitter Engine to wait conditionally until a given synchronization register gets updated with a value greater than the

"SEMAPHORE_DATA_DWORD" mentioned inline in this command. Synchronization registers can be updated through CPU MMIO access or through execution of MI_LOAD_REGISTER_IMM command in other engines. Synchronization between contexts (especially between contexts running on 2 different engines) is provided by the MI_SEMAPHORE_MBOX command.

If execution is stalled due to this command, the engine will specify that the engine is IDLE to the power management engine.

DWord	Bit	Description							
0	31:29	Command Type							
		Default Value:			0h MI_C	COMMAND			
		Format:			OpCode	е			
	28:23	MI Command Opcode							
		Default Value: 16h MI_SEMAPI			SEMAPH	IORE_M	BOX		
		Format:		OpCode					
	22:21	Reserved							
		Format:					MBZ		
	20	Reserved							
		Default Value:				1h			
		Format:				Must Be One			
	19	Reserved							
		Format:					MBZ		
	18	Reserved							
		Default Value:				lh			
		Format:				Must Be One			
	17:16	Register Select This field indicates the s	synchroniz	zation reg	gister to	be used	I for comparison with the inline data.		
		Value				Name			
		0h	CS regist	er (BRSYI	NC)				
		1h	Reserved						
		2h	VCS regi	ser (BVSY	NC)				
		3h	Reserved						
	15:8	Reserved							
		Format:					MBZ		
	7:0	DWord Length							



	MI_SEMAPHORE_MBOX						
		Default Value:	1h Excludes DWord (0,1)				
		Format:	=n Total Length - 2				
1	31:0	S1:0 Semaphore Data Dword					
		Format: U32					
		Inline Data Dword to compare with the selected synchronization register. The Data dword is supplied by software to control execution of the command buffer. If the data in the selected synchronization register is greater than this dword, the execution of the command buffer continues.					
2	31:0	Reserved					
		Format:		MBZ			



MI_SEMAPHORE_MBOX

Source: RenderCS

2

Length Bias:

MI_SEMAPHORE_MBOX command provides capability in Render Engine to wait conditionally until a given synchronization register gets updated with a value greater than the "SEMAPHORE_DATA_DWORD" mentioned inline in this command. Synchronization registers can be updated through CPU MMIO access or through execution of MI_LOAD_REGISTER_IMM command in other engines.

DWord	Bit	Description							
0	31:29	Command Type			1				
		Default Value:			0h MI_C		ND		
		Format:			OpCode	е			
	28:23	MI Command Opcod	e						
		Default Value:		16h MI_	SEMAPH	IORE_M	BOX		
		Format: OpCode)				
	22:21	Reserved							
		Format:				MBZ			
	20	Reserved							
		Default Value:			1h				
		Format:				Must B	e One		
	19	Reserved							
		Format:					MBZ		
	18	Reserved				<u>. </u>			
		Default Value:			1h				
		Format:				Must Be One			
	17:16	Register Select			•				
		This field indicates the	synchroni	zation reg	gister to	b be used for comparison with the inline data.			
		Oh	DVSVNC	Name		Description			
		011 1.h	RUSTINC			VCS Register			
		2h				BC	S Register		
		3h	Reserved			Red	served		
	15.1/	Posorvod	Reserved						
	13.14	Format:					МВZ		
	13.8	Reserved					<u> </u>		
	10.0	Format:					MBZ		
	7:0	DWord Length					<u></u>		
		Default Value:	1h						
		Format:	=n Tot	al Length	- 2. Excl	udes D	Word (0,1).		
1	31:0	Semaphore Data Dwo	ord						



MI_SEMAPHORE_MBOX

		Format: Inline Data Dword to compare with the selected synchroniz supplied by software to control execution of the command synchronization register is greater than this dword, the exec continues.	U32 tation register. The Data dword is buffer. If the data in the selected cution of the command buffer
2	31:0	Reserved Format:	MBZ



Source:

Length Bias:

VideoCS

2

Description

MI_SEMAPHORE_MBOX command provides capability in Video Engine to wait conditionally until a given synchronization register gets updated with a value greater than the "SEMAPHORE_DATA_DWORD" mentioned inline in this command. Synchronization registers can be updated through CPU MMIO access or through execution of MI_LOAD_REGISTER_IMM command in

other engines.

If execution is stalled due to this command, the engine will specify that the engine is IDLE to the power management engine.

DWord	Bit				Descrip	tion		
0	31:29	Command Type						
		Default Value:			0h MI_C	1I_COMMAND		
		Format:		OpCoo				
	28:23	MI Command Opcode	9					
		Default Value:		16h MI_	SEMAPH	ORE_MI	BOX	
		Format:		OpCode	i.			
	22:21	Reserved						
		Format:					MBZ	
	20	Reserved	served					
		Default Value:				1h		
		Format:				Must Be One		
	19	Reserved						
		Format:				MBZ		
	18	Reserved						
		Default Value:				1h		
		Format:				Must Be One		
	17:16	Register Select						
		This field indicates the	synchroniz	zation reg	gister to	be used for comparison with the inline data.		
		Value				Name		
		0h	BCS regis	ter (VBSY	'NC)			
		1h	Reserved					
		2h	CS registe	er (VRSYN	IC)			
		3h	Reserved					
	15:8	Reserved						
		Format:					MBZ	
	7:0	DWord Length						
		Default Value:		1h Ex	cludes D	ludes DWord (0,1)		



MI	SEM	APH	ORE	MBOX
_	-		_	-

		Format:	=n Total Length - 2	
1	31:0	Semaphore Data Dword		
		Format:		U32
		Inline Data Dword to compare with the selected synchronization register. The Data dword is supplied by software to control execution of the command buffer. If the data in the selected synchronization register is greater than this dword, the execution of the command buffer continues.		
2	31:0	Reserved		
		Format:		MBZ



MI_SET_CONTEXT

Source: RenderCS Length Bias: 2

The MI_SET_CONTEXT command is used to specify the *logical* context associated with the hardware context. A logical context is an area in memory used to store hardware context information, and the context is referenced via a 2KB-aligned pointer. If the (new) logical context is different (i.e., at a different memory address), the device saves the current HW context values to the current logical context address, and then restores (loads) the new logical context by reading the context from the new address and loading it into the hardware context state. If the logical context address specified in this command matches the current logical context address, this command is effectively treated as a NOOP. **Specific to the Render command stream only.**

This command also includes some controls over the context save/restore process.

- The **Force Restore** bit can be used to refresh the on-chip device state from the same memory address if the indirect state buffers have been modified.
- The **Restore Inhibit** bit can be used to prevent the new context from being loaded at all. This must be used to prevent an uninitialized context from being loaded. Once software has initialized a context (by setting all state variables to initial values via commands), the context can then be stored and restored normally.
- This command needs to be always followed by a single MI_NOOP instruction to workaround a silicon issue.
- When switching from a generic media context to a 3D context, the generic media state must be cleared via the Generic Media State Clear bit 16 in PIPE_CONTROL (or bit 4 in MI_FLUSH) before saving 3D context.
- MI_SET_CONTEXT commands are permitted only within a ring buffer (not within a batch buffer).

Programming Notes

MI_ARB_ON_OFF with 'Arbitration Enable Reset' set should be programmed before an MI_SET_CONTEXT command. MI_ARB_ON_OFF with 'Arbitration Enable' set should be programmed after an MI_SET_CONTEXT command. This programming ensures that PSMI context switch flows do not conflict with MI_SET_CONTEXT flows.

DWord	Bit			Description		
0	31:29	Command Type				
		Default Value:		0h MI_COMMA	0h MI_COMMAND	
		Format:		OpCode	OpCode	
	28:23	MI Command Opcode				
		Default Value:		L8h MI_SET_CONTEXT		
		Format:		0pCode		
	22:8	Reserved				
		Format:			MBZ	
	7:0	DWord Length				
		Default Value:	0h			
		Format:	=n Total Len	gth - 2. Excludes DWord (0,1).		



			MI_SE	CONTEXT					
1	31:12	Logical Context Address							
		Format:	GraphicsAddress[31:12]LogicalContext					
		Description							
		This field co	This field contains the 4KB-aligned graphics memory address of the Logical						
		Context that is to be loaded into the hardware context. If this address is equal							
		to the CCID register associated with the current ring, no load will occur. Prior							
		to loading th	his new context, the	device will save the	existing context as				
		required. Aft	ter the context switcl	n operation comple	tes, this address will be				
				introduced					
-			eas to be 4KB alighed v						
	11:10	Reserved							
		Format:			MBZ				
	9	Reserved							
		Format:			MBZ				
	8	Reserved, Mu	ust be 1						
		Format:		Must Be One					
	7:5	Reserved							
		Format:			MBZ				
	4	Reserved			1				
		Format:			MBZ				
	3	Extended Stat	te Save Enable						
		Format:		Enabl	e				
		It 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. This bit will be							
		stored in the associated CCID register to control the context save operation when switchir							
		from this context (as part of a subsequent MI_SET_CONTEXT command). This bit must be 1 when							
		RS2 power sta	ate is enabled (via MCH	IBAR, offset 0x11B8)					
	2	Extended Stat	te Restore Fnable						
	2	Format:		Enabl	<u>م</u>				
		If set, the extended state identified in the Logical Context Data section of the Memon							
		If set, the exte	ended state identified i	in the Logical Context	Data section of the Memory Data				
		If set, the exter Formats chapt	ended state identified i ter is loaded (or restore	in the Logical Context ed) as part of switchin	Data section of the Memory Data g to this logical context. This method				
		If set, the externation of the set, the externation of the set of	ended state identified i ter is loaded (or restore o restore things such a	in the Logical Context ed) as part of switchin s filter coefficients usi	Data section of the Memory Data g to this logical context. This method ng the indirect state restore followed				
		If set, the externation Formats chapt can be used to by a restore of	ended state identified i ter is loaded (or restore o restore things such as f the extended logical	in the Logical Context ed) as part of switchin s filter coefficients usi context data. This bit	Data section of the Memory Data g to this logical context. This method ng the indirect state restore followed affects the switch (if required) to the				
		If set, the exter Formats chapt can be used to by a restore of context specifi register to con	ended state identified i ter is loaded (or restore o restore things such as f the extended logical fied in Logical Context a ntrol a subsequent con	in the Logical Context ed) as part of switchin s filter coefficients usi context data. This bit Address. This bit will a text save operation w	Data section of the Memory Data g to this logical context. This method ng the indirect state restore followed affects the switch (if required) to the also be stored in the associated CCID hen switching to this context (as part of				
		If set, the exter Formats chapt can be used to by a restore of context specifi register to con a subsequent	ended state identified i ter is loaded (or restore o restore things such a f the extended logical fied in Logical Context a ntrol a subsequent con ring buffer switch). Thi	in the Logical Context ed) as part of switchin s filter coefficients usi context data. This bit Address. This bit will a text save operation w s bit must be 1 when	Data section of the Memory Data g to this logical context. This method ng the indirect state restore followed affects the switch (if required) to the also be stored in the associated CCID hen switching to this context (as part of RS2 power state is enabled (via				
		If set, the externation Formats chapt can be used to by a restore of context specifi register to con a subsequent MCHBAR, offse	ended state identified i ter is loaded (or restore o restore things such a f the extended logical fied in Logical Context ntrol a subsequent con ring buffer switch). Thi set 0x11B8)	in the Logical Context ed) as part of switchin s filter coefficients usi context data. This bit Address. This bit will a text save operation w s bit must be 1 when	Data section of the Memory Data g to this logical context. This method ng the indirect state restore followed affects the switch (if required) to the also be stored in the associated CCID hen switching to this context (as part of RS2 power state is enabled (via				
	1	If set, the exter Formats chapt can be used to by a restore of context specifi register to con a subsequent MCHBAR, offse	ended state identified i ter is loaded (or restore o restore things such a f the extended logical fied in Logical Context ntrol a subsequent con ring buffer switch). Thi set 0x11B8)	in the Logical Context ed) as part of switchin s filter coefficients usi context data. This bit Address. This bit will a text save operation w s bit must be 1 when	Data section of the Memory Data g to this logical context. This method ng the indirect state restore followed affects the switch (if required) to the also be stored in the associated CCID hen switching to this context (as part of RS2 power state is enabled (via				

Г



MI_SET_CONTEXT							
	contests of the CCID register is performed. Normally, matching addresses prevent a context restore from occurring; however, when this bit is set a context restore is forced to occur. This bit cannot be set with Restore Inhibit. Note: This bit is not saved in the associated CCID register. It only affects the processing of this command.						
0	Restore Inhibit If set, the restore of the HW context from the logical context specified by Logical Context Address is inhibited (i.e., the existing HW context values are maintained). This bit must be used to prevent the loading of an uninitialized logical context. If clear, the context switch proceeds normally. This bit cannot be set with Force Restore. Note: This bit is not saved in the associated CCID register. It only affects the processing of this command.						



Source: RenderCS

2

Length Bias:

1

I.

The MI_STORE_DATA_IMM command requests a write of the QWord constant supplied in the packet to the specified Memory Address. As the write targets a System Memory Address, the write operation is coherent with the CPU cache (i.e., the processor cache is snooped).

Programming Notes

- This command can be used for general software synchronization through variables in cacheable memory (i.e., where software does not need to poll un-cached memory or device registers).
- This command simply initiates the write operation with command execution proceeding normally. Although the write operation is guaranteed to complete eventually, there is no mechanism to synchronize command execution with the completion (or even initiation) of these operations.

DWord	Bit			Description			
0	31:29	Command Type					
		Default Value:		0h MI_COMMA	ND		
		Format:		OpCode			
	28:23	MI Command Opcode					
		Default Value:	20h M	_STORE_DATA_I	MM		
		Format:	OpCoc	le			
	22	Use Global GTT					
		Format:		Boolean			
If set, this command will use the global GTT to translate the Address. If clear, the used. This bit must be '1' if the Per Process GTT Enable bit is clear.							
	21	Reserved					
		Format:			MBZ		
	20:10	Reserved					
		Format:			MBZ		
	9:6	Reserved					
		Format:			MBZ		
	5:0	DWord Length					
		Default Value:	2h Excludes DW	/ord (0,1)			
		Format:	Format: = n Total Length - 2. Excludes DWord (0,1)				
			Due				
		Programming Notes					
		Dword Length programi	th programmed must not exceed 0x3.				
1	31:0	Reserved			MDZ		
		Format:			IVIBZ		
2	31:2	Address					



		Format:	GraphicsAddress[31:2]U32(2)						
		This field specifies Bit address must be DWc for a store "QW" com	s 31:2 of the Address where the DWord will be stored. As the store rd-aligned, Bits 1:0 of that address MBZ. This address must be 8B aligned nand.						
	1:0	Reserved	Reserved						
		Format:		MBZ					
3	31:0	Data DWord 0							
		Format:		U32					
		This field specifies the this DWord is the lowe	the DWord value to be written to the targeted location. For a QWord write wer DWord of the QWord to be reported (DW 0).						
4	31:0	Data DWord 1							
		U32							
		This field specifies the upper DWord value to be written to the targeted QWord location (D 1).							



Source: BlitterCS

2

Length Bias:

The MI_STORE_DATA_IMM command requests a write of the QWord constant supplied in the packet to the specified Memory Address. As the write targets a System Memory Address, the write operation is coherent with the CPU cache (i.e., the processor cache is snooped).

Programming Notes

This command can be used for general software synchronization through variables in cacheable memory (i.e., where software does not need to poll un-cached memory or device registers). However, the cacheable nature of the transaction is determined by the setting of the "mapping type" in the GTT entry. This command simply initiates the write operation with command execution proceeding normally. Although the write operation is guaranteed to complete "eventually", there is no mechanism to synchronize command execution with the completion (or even initiation) of these operations. All writes to memory generated using this command are expected to finish in order.

DWord	Bit	Description					
0	31:29	Comma	and Type				
		Defaul	t Value:		0h MI_COMMA	AND	
	28:23	MI Cor	MI Command Opcode				
		Defaul	t Value:		20h MI_STORE_DATA_I	IMM	
	22	Use Glo	obal GTT				
		This bit	must be '1' if	the Per Pro	cess GTT Enable bit is cle	ear.	
		Value	Nan	ne	Des	scription	
		0h	Per Process G Address	Graphics	This command will use t translate the Address	the per process GTT to	
1hGlobal Graphics1Addresst			iics	This command will use t the Address.	the global GTT to translate		
	21	Reserved					
		Forma	t:			MBZ	
	20:10	Reserved					
		Format: MBZ				MBZ	
	9:0	DWord	l Length				
		Defaul	t Value:	2h Exclude	des DWord (0,1) = 2 for DWord, 3 for QWord		
		Forma	t:	=n Total Le	ength - 2		
1	31:0	Reserv	ed				
		Forma	t:			MBZ	
2	31:2 Address						
	Format: GraphicsAddress[31:2]U32(2)						
This field specifies Bits 31:2 of the Address where the DWord will be stored. As the address must be DWord-aligned, Bits 1:0 of that address MBZ. This address must for a store "QW" command.						ord will be stored. As the store MBZ. This address must be 8B aligned	



	MI_STORE_DATA_IMM							
	1:0	Reserved						
		Format:	MBZ					
3	31:0	Data DWord 0						
		Format:	U32					
		This field specifies the DWord value to be written to the ta this DWord is the lower DWord of the QWord to be report	s field specifies the DWord value to be written to the targeted location. For a QWord write DWord is the lower DWord of the QWord to be reported (DW 0).					
4	31:0	Data DWord 1						
		Format:	U32					
		This field specifies the upper DWord value to be written to the targeted QWord location (DW 1).						



Source: VideoCS

2

Length Bias:

The MI_STORE_DATA_IMM command requests a write of the QWord or DWord constant supplied in the packet to the specified Memory Address. As the write targets a System Memory Address, the write operation is coherent with the CPU cache (i.e., the processor cache is snooped).

Programming Notes

This command can be used for general software synchronization through variables in cacheable memory (i.e., where software does not need to poll un-cached memory or device registers).

This command simply initiates the write operation with command execution proceeding normally. Although the write operation is guaranteed to complete "eventually", there is no mechanism to synchronize command execution with the completion (or even initiation) of these operations.

DWord	Bit	Description					
0	31:29	Command Type					
		Default Value:			0h MI_COMMAND		
		Format:			OpCode		
	28:23	MI Command Opco	de				
		Default Value:		20h MI	STORE_DATA_I	MM	
		Format:		OpCod	le		
	22	Use Global GTT					
		Format:					U1
		If set, this command	will use the g	Ilobal G	TT to translate th	ne Addre	ess. If clear, the PPGTT will be
		used. This bit must b	e '1' if the Pe	r Proces	s GTT Enable bit	is clear.	
	21.0	Pacanyod					
	21.0	Format:				MB7	
	7:0	DWord Length					
		Default Value:	0h Excludes	DWord	(0.1) = 3 for OW	lord 2 f	or DWord
		Format:	=n Total Ler	hath - 2	(0,2) 0 101 Q11		
1	31.0	Reserved		<u> </u>			
T	51.0	Format [.]				MB7	
2	21.2	Address					
Z	51.2	Format:	Graphi	cc Addra	acc[31·2]		
		This field specifies Bit	Giapin	Addroc	= 55[JI.2]	محطيبنالا	a stared As the stare address
		must be DWord-aligned. Bits 1:0 of that address MB7. This address must be SB aligned for a					
		store "OW" command					
			G.				
	1:0	Reserved					
		Format:				MBZ	
3	31:0	Data DWord 0					



	MI_STORE_DATA_IMM							
		Format: U32 FormatDesc						
		This field specifies the DWord va this DWord is the lower DWord	rd value to be written to the targeted location. For a QWord write ord of the QWord to be reported (DW 0).					
4	31:0	Data DWord 1						
		Format:	U32 FormatDesc					
		Vord value to be written to the targeted QWord location (DW 1).						



Source: BlitterCS

2

Length Bias:

The MI_STORE_DATA_INDEX command requests a write of the data constant supplied in the packet to the specified offset from the System Address defined by the Hardware Status Page Address Register. As the write targets a System Address, the write operation is coherent with the CPU cache (i.e., the processor cache is snooped).

Programming Notes

Use of this command with an invalid or uninitialized value in the Hardware Status Page Address Register is UNDEFINED. This command can be used for general software synchronization through variables in cacheable memory (i.e., where software does not need to poll uncached memory or device registers). This command simply initiates the write operation with command execution proceeding normally. Although the write operation is guaranteed to complete "eventually", there is no mechanism to synchronize command execution with the completion (or even initiation) of these operations.

DWord	Bit	Description						
0	31:29	Command Ty	Command Type					
		Default Value: 0h MI_COMMAND						
	28:23	MI Comman	d Opco	de				
		Default Value	e:	21h MI_STORE_DA	TA_IN	IDEX		
	22	Reserved						
		Format:				MBZ		
	21	Reserved						
		Format:				MBZ		
	20:8	Reserved	Reserved					
		Format:				MBZ		
	7:0	DWord Leng	DWord Length					
		Default Value	e:	1h Excludes DWord $(0,1) = 1$ for	or DV	Vord, 2 for QWord		
		Format:		=n Total Length - 2				
1	31:12	Reserved						
		Format:				MBZ		
	11:2	Offset						
		Format:	U10 ze	ro-based DWord offset into the	HW	status page.		
		Format:	Hardwa	reStatusPageOffset[11:2]U32				
		This field spe	cifies th	e offset (into the hardware statu	ıs pag	ge) to which the data will be written.		
		Note that the	first fev	v DWords of this status page ar	e rese	erved for special-purpose data storage		
- targeting these reserved locations via this command is UNDEFINED. This addre								
		Value Name						
		[16, 1023]						
	1:0	Reserved						



MI_STORE_DATA_INDEX								
		Format: MBZ						
2	31:0	Data DWord 0						
		Format:	U32					
		This field specifies the DWord value to be written to the tai	geted location. For a QWord write					
		this DWord is the lower DWord of the QWord to be reported (DW 0).						
3	31:0	Data DWord 1						
		Format:	U32					
		This field specifies the upper DWord value to be written to	the targeted QWord location (DW 1).					



Source: RenderCS

2

Length Bias:

The MI_STORE_DATA_INDEX command requests a write of the data constant supplied in the packet to the specified offset from the System Address defined by the Hardware Status Page Address Register. As the write targets a System Address, the write operation is coherent with the CPU cache (i.e., the processor cache is snooped).

Programming Notes

- Use of this command with an invalid or uninitialized value in the Hardware Status Page Address Register is UNDEFINED.
- This command can be used for general software synchronization through variables in cacheable memory (i.e., where software does not need to poll uncached memory or device registers).
- This command simply initiates the write operation with command execution proceeding normally. Although the write operation is guaranteed to complete eventually, there is no mechanism to synchronize command execution with the completion (or even initiation) of these operations.

DWord	Bit	Description				
0	31:29	Command Ty	pe			
		Default Value: 0h MI_COMN				ND
		Format:			OpCode	
	28:23	MI Comman	d Opcode			
		Default Value	2:	21h MI_	STORE_DATA_IN	DEX
		Format:		OpCode		
	22	Reserved				
	21	Reserved				
		Format:				MBZ
	20:8	Reserved				
		Format:				MBZ
	7:0	DWord Leng	th			
		Default Value	e: 1h			
		Format:	=n Total Length	n - 2. Excl	udes DWord (0,2	1) = 1 for DWord, 2 for QWord.
1	31:12	Reserved				
		Format:				MBZ
	11:2	Offset				
		Format:	U10 zero-based D	Nord offs	et into the HW s	status page.
	Format: HardwareStatusPageOffset[11:2]U32					
		This field specifies the offset (into the hardware status page) to which the data will be written				
		Note that the	first few DWords o	f this stat	us page are rese	erved for special-purpose data storage
		- targeting th	ese reserved locatio	ons via thi	s command is U	NDEFINED. This address must be 8B
		aligned for a	store Qw command	J.		



	MI_STORE_DATA_INDEX									
		Value	Name							
		[16, 1023]								
	1:0	Reserved	Reserved							
		Format:	MBZ							
2	31:0	Data DWord 0								
		Format:	U32							
		This field specifies the DWord value to be written to the targeted location. For a QWord write this DWord is the lower DWord of the QWord to be reported (DW 0).								
3	31:0	Data DWord 1								
		Format:	U32							
		This field specifies the upper DWord value to be written to	o the targeted QWord location (DW 1).							



Source: VideoCS

2

Length Bias:

The MI_STORE_DATA_INDEX command requests a write of the data constant supplied in the packet to the specified offset from the System Address defined by the Hardware Status Page Address Register. As the write targets a System Address, the write operation is coherent with the CPU cache (i.e., the processor cache is snooped).

Programming Notes

- Use of this command with an invalid or uninitialized value in the Hardware Status Page Address Register is UNDEFINED.
- This command can be used for general software synchronization through variables in cacheable memory (i.e., where software does not need to poll uncached memory or device registers).
- This command simply initiates the write operation with command execution proceeding normally. Although the write operation is guaranteed to complete "eventually", there is no mechanism to synchronize command execution with the completion (or even initiation) of these operations.

DWord	Bit					Description		
0	31:29	Command T	уре					
		Default Valu	ie:			0h MI_COMM	AND	
		Format:				OpCode		
	28:23	MI Comman	d Opcode					
		Default Valu	ie:		21h MI_	STORE_DATA_II	NDEX	
		Format:			OpCode			
	22	Reserved						
		Format:					MBZ	
	21	Reserved						
		Format:					MBZ	
20:8 Reserved								
		Format:					MBZ	
	7:0	DWord Length						
		Default Valu	ie:	0h Ex	cludes D\	Word (0,1) = 2 f	rd (0,1) = 2 for QWord	
		Format:		=n To	otal Lengt	th - 2		
1	31:12	Reserved						
		Format:					MBZ	
	11:2	Offset	1					
		Format:	U10 zero-ba	ased D	Word off	set into the HW	/ status page	
		Format: GraphicsAddress[11:2]U32						
		This field spe	ecifies the off	set (in Iffset is	to the ha	rdware status p wn to bit 3 only	age) to which the data will be written.	
				Value			Name	



		_					
		[16, 1023]					
		Programming Notes					
		The first few DWords of this status page are reserved for special-purpose data storage -					
		targeting these reserved locations via this command is UNDEFINED.					
	1:0	Reserved					
		Format:		MBZ			
2	31:0	Data DWord 0					
		Format:	U32 FormatDesc				
		This field specifies the upper DWord value to be written to the targeted QWord location (DW					
		1).					
3	31:0	Data Word 1					
		Format:	U32 FormatDesc				
	This field specifies the upper DWord value to be written to the targeted QWord lo						
		1).					



MI_STORE_REGISTER_MEM

Source: CommandStreamer

Length Bias:

2

The MI_STORE_REGISTER_MEM command requests a register read from a specified memory mapped register location in the device and store of that DWord to memory. The register address is specified along with the command to perform the read.

Programming Notes

- The command temporarily halts command execution.
- The memory address for the write is snooped on the host bus.
- This command will cause undefined data to be written to memory if given register addresses for the PGTBL_CTL_0 or FENCE registers.

DWord	Bit	Description							
0	31:29	Command Type							
		Default Value:			0h MI_COMMAND				
		Forma	t:				OpCode		
	28:23	MI Command Opcode							
		Default Value: 24			24h	h MI_STORE_REGISTER_MEM			
		Format:			OpCode				
	22	Use Global GTT This bit must be 1 if the Per Process GTT Enable bit is clear.							
		Value	ue Name			Description			
		0h	Per Process Graphics Address		-	This command will use the per process GTT to translate the Address.			
		1h	Global Graphics Address		s ,	This command will use the global GTT to translate the Address.			
	21	Reserved							
		Format:						MBZ	
	20:8	Reserve	Reserved						
		Format:						MBZ	
	7:0	DWord Length							
		Default Value:				1h Excludes DWord (0,1)			
		Format:				=n Total Length - 2			
1	31:23 Reserved								
		Format:					MBZ		
	22:2	Register Address							
		Forma	t:	MMIOAddre	ess[2	2:2]MI	MIO_Register		
		This field specifies Bits 22:2 of the Register offset the DWord will be read from. As the register address must be DWord-aligned, Bits 1:0 of that address MBZ.							

г



MI_STORE_REGISTER_MEM							
		Programming Notes					
		 Storing a V The values UNDEFINE specified. 	VGA register is not permitted and will store an UNDEFINED value. es of PGTBL_CTL0 or any of the FENCE registers cannot be stored to memory; NED values will be written to memory if the addresses of these registers are				
	1:0	Reserved					
		Format:		MBZ			
2	31:2 Memory Address						
		Format:	GraphicsAddress[31:2]MMIO_Register				
		This field specifies the address of the memory location where the register value specified in the DWord above will be written. The address specifies the DWord location of the data. Range = GraphicsVirtualAddress[31:2] for a DWord register					
	1:0	Reserved					
		Format:		MBZ			


MI_TOPOLOGY_FILTER

Source:		RenderCS			
Length Bia	is:	1			
This comm command command	nand is u s that do with a T	used to specify a specific 3DPrin to no have a matching 3DPrimTy Fopology Filter Value of 0). This	nType valu vpe. This p comman	ie, where the CS rimitive culling i d is specific to	will ignore all 3DPRIMITIVE s optional (turned off by using this the Render command stream only.
DWord	Bit			Description	
0	31:29	Command Type			
		Default Value:		0h MI_COMMA	ND
		Format:		OpCode	
	28:23	MI Command Opcode			
		Default Value:	0Dh MI	_TOPOLOGY_FIL	TER
		Format:	OpCod	e	
-	22:6	Reserved			
		Format:			MBZ
-	5:0	Topology Filter Value			
		Format:	3D_PrimTo	роТуре	
		When non-zero, the CS will disc specified 3DPrimTopologyType	card all 3D . When ze	PRIMITIVE comi ro, no filtering is	mands which do not match the s performed (normal operation).



MI_URB_CLEAR

Source: RenderCS

Length Bias:

The MI_URB_CLEAR command allows SW to clear (write zero) to a section in the URB.

Programming Notes

- The command temporarily halts command execution.
- This command is part of context save/restore. Only the last instance will be part of context.
- This command requires the 3D pipeline to be flushed before execution.

MI_URB_CLEAR must be programmed following MI_SET_CONTEXT and before workload is submitted, when a given context expects URB locations to be initialized to 0x0.

DWord	Bit			Description	
0	31:29	Command Type			
		Default Value:		0h MI_COMM	IAND
		Format:		OpCode	
	28:23	MI Command Opcode			
		Default Value:		19h MI_URB_C	LEAR
		Format:		OpCode	
	22:8	Reserved			
		Format:			MBZ
	7:0	DWord Length			1
		Default Value:	0h		
		Format:	=n Total Lengt	th - 2. Excludes	DWord (0,1).
1	31:30	Reserved			
		Format:			MBZ
	29	Reserved			
		Format:			MBZ
	28:16	URB Clear Length This field specifies the	number of 256k	o entries in the	URB to be cleared to zero.
		V	alue		Name
		[0,8191]			
	15	Reserved			
		Format:			MBZ
	14	Reserved			
		Format:			MBZ
	13:0	URB Address			
		Format: UR	BAddress[18:5]	256b aligned	
		This field specifies Bits	18:5 of the URB	8 Address	



MI_USER_INTERRUPT

Source: Length Bias:		BlitterCS 1			
The MI_USE	R_INTERR r processi	UPT command is used to generate ng this command. See User Intern	e a Use upt.	r Interrupt con	dition. The parser will continue
DWord	Bit			Description	
0	31:29	Command Type			
		Default Value:		0h MI_COMMA	AND
	28:23	MI Command Opcode			
		Default Value:	02h N	1I_USER_INTERF	RUPT
	22:0	Reserved			
		Format:			MBZ



		MI_USER_I	NTER	RUPT	
Source:	RenderC	S			
Length Bias:	1				
The MI_USER_IN parsing after proc	TERRUPT com cessing this co	imand is used to generate ommand. See User Interrup	a User Int ot.	errupt conditi	ion. The parser will continue
DWord	Bit			Description	
0	31:29	Command Type			
		Default Value:		0h MI_COM	MAND
		Format:		OpCode	
	28:23	MI Command Opcode			
		Default Value:	02h	MI_USER_INTE	ERRUPT
		Format:	OpC	ode	
	22:0	Reserved			
		Format:			MBZ



MI_USER_INTERRUPT

VideoCS				
1				
ERRUPT comr essing this cc	mand is used to generate a U ommand. See User Interrupt.	ser Inte	errupt conditio	on. The parser will continue
Bit		l I	Description	
31:29	Command Type			
	Default Value:		0h MI_COMN	MAND
	Format:		OpCode	
28:23	MI Command Opcode			
	Default Value:	02h I	MI_USER_INTE	RRUPT
	Format:	ОрСо	ode	
22:0	Reserved			
	Format:			MBZ
	VideoCS 1 ERRUPT commensions of the second s	VideoCS 1 ERRUPT command is used to generate a U essing this command. See User Interrupt. Bit 31:29 Command Type Default Value: Format: 28:23 MI Command Opcode Default Value: Format: 22:0 Reserved Format:	VideoCS 1 ERRUPT command is used to generate a User Interessing this command. See User Interrupt. Bit Command Type Default Value: Format: 28:23 MI Command Opcode Default Value: format: 22:0 Reserved Format:	VideoCS 1 ERRUPT command is used to generate a User Interrupt conditionessing this command. See User Interrupt. Bit Description 31:29 Command Type Default Value: 0h MI_COMN Format: 0pCode 28:23 MI Command Opcode Default Value: 02h MI_USER_INTE Format: 0pCode 22:0 Reserved Format:



MI_WAIT_FOR_EVENT

Source: BlitterCS

Length Bias:

The MI_WAIT_FOR_EVENT command is used to pause command stream processing until a specific event occurs or while a specific condition exists. Only one event/condition can be specified -- specifying multiple events is UNDEFINED. The effect of the wait operation depends on the source of the command. If executed from a batch buffer, the parser will halt (and suspend command arbitration) until the event/condition occurs. If executed from a ring buffer, further processing of that ring will be suspended, although command arbitration (from other rings) will continue. Note that if a specified condition does not exist (the condition code is inactive) at the time the parser executes this command, the parser proceeds, treating this command as a no-operation. If execution of this command from a primary ring buffer causes a wait to occur, the active ring buffer will effectively give up the remainder of its time slice (required in order to enable arbitration from other primary ring buffers).

DWord	Bit				Description		
0	31:29	Commar	nd Type				
		Default	Value:		0h MI_COM	MAND	
	28:23	MI Com	mand Opcode	I.			
		Default	Value:		03h MI_WAIT_FOR_	EVENT	
	22	Reserved	d				
		Format:				MBZ	
	21	Reserved	d				
		Format:				MBZ	
	20	Display S	Sprite C Flip P	ending Wa	it Enable		
		Format:			Ena	ble	
		request i buffer ad	s pending, the Idress has now	parser will been loade	wait until the flip ope ed into the active fro	eration has completed (i.e., the non- nt buffer registers).	ew front
	19:16	Conditio This field enable se that cond	on Code Wait S enables a wait elect one of 15 dition-code in t	Select t for the du condition o the EXCC is	ration that the corres codes in the EXCC re- cleared.	sponding condition code is active gister, that cause the parser to w	e. These ait until
		Value	Name		Descr	iption	
		0h	Not Enabled	Condition	Code Wait not enabl	ed	
		1h-5h	Enabled	Condition	Code select enabled;	selects one of 5 codes, 0 - 4	
		6h-15h	Reserved				
					Programming No	otes	
		Note that	at not all condi	tion codes	are implemented. Th	e parser operation is UNDEFINE	D if an
		(Memor	y Interface Reg	jisters) lists	the codes that are in	nplemented.	gister
	15	Display	Plane C Flip Po	ending Wa	it Enable		
		Format:			Ena	ble	
		This field	enables a wait	t for the du	ration of a Display Pl	ane C "Flip Pending" condition. I	f a flip



	request is pending, the parser will wait until th buffer address has now been loaded into the a	e flip operation has completed (i.e., the new active front buffer registers).
14	Reserved	
	Format:	MBZ
13:12	Reserved	
	Format:	MBZ
11	Reserved	
	Format:	MBZ
10	Display Sprite B Flip Pending Wait Enable	
	Format:	Enable
	request is pending, the parser will wait until th buffer address has now been loaded into the a	e flip operation has completed (i.e., the new active front buffer registers).
9	Display Plane B Flip Pending Wait Enable	
	Format:	Enable
	i filis lielu enables a walt for the duration of a t	Display Plane B "Flip Pending" condition. If a
8	request is pending, the parser will wait until th buffer address has now been loaded into the a	Display Plane B "Flip Pending" condition. If a le flip operation has completed (i.e., the new active front buffer registers).
8	request is pending, the parser will wait until th buffer address has now been loaded into the a Reserved Format:	Display Plane B "Flip Pending" condition. If a le flip operation has completed (i.e., the new active front buffer registers). MBZ
8	Reserved Reserved	Display Plane B "Flip Pending" condition. If a ne flip operation has completed (i.e., the new front buffer registers).
8 7:6	Reserved Format:	Display Plane B "Flip Pending" condition. If a ne flip operation has completed (i.e., the new front buffer registers). MBZ
8 7:6 5:4	Reserved Format: Reserved Format: Reserved	Display Plane B "Flip Pending" condition. If a le flip operation has completed (i.e., the new active front buffer registers). MBZ MBZ
8 7:6 5:4	Reserved Format: Reserved Format: Reserved Format:	Display Plane B "Flip Pending" condition. If a ne flip operation has completed (i.e., the new active front buffer registers). MBZ MBZ
8 7:6 5:4	Reserved Format:	Display Plane B "Flip Pending" condition. If a ne flip operation has completed (i.e., the new front buffer registers). MBZ MBZ MBZ
8 7:6 5:4 3	Reserved Format:	Display Plane B "Flip Pending" condition. If a le flip operation has completed (i.e., the new active front buffer registers). MBZ MBZ MBZ
8 7:6 5:4 3	Reserved Format: Reserved Format: Reserved Format: Reserved Format: Reserved Format: Display Sprite & Flip Pending Wait Enable	Display Plane B "Flip Pending" condition. If a le flip operation has completed (i.e., the new active front buffer registers). MBZ MBZ MBZ MBZ
8 7:6 5:4 3 2	Reserved Format: Reserved Format: Reserved Format: Reserved Format: Display Sprite A Flip Pending Wait Enable Format:	Display Plane B "Flip Pending" condition. If a ne flip operation has completed (i.e., the new front buffer registers). MBZ MBZ MBZ MBZ Enable
8 7:6 5:4 3 2	Reserved Format: This field enables a wait for the duration of a D request is pending, the parser will wait until the buffer address has now been loaded into the addres	Display Plane B "Flip Pending" condition. If a ne flip operation has completed (i.e., the new front buffer registers). MBZ MBZ MBZ MBZ Enable Display Sprite A "Flip Pending" condition. If a ne flip operation has completed (i.e., the new front buffer registers).
8 7:6 5:4 3 2	Inits field enables a wait for the duration of a trequest is pending, the parser will wait until the buffer address has now been loaded into the address has now been	Display Plane B "Flip Pending" condition. If a ne flip operation has completed (i.e., the new active front buffer registers). MBZ MBZ MBZ Bable Display Sprite A "Flip Pending" condition. If a ne flip operation has completed (i.e., the new active front buffer registers).
8 7:6 5:4 3 2	Inits field enables a wait for the duration of a trequest is pending, the parser will wait until the buffer address has now been loaded into the address has now been	Display Plane B "Flip Pending" condition. If a ne flip operation has completed (i.e., the new front buffer registers). MBZ MBZ MBZ MBZ Enable Display Sprite A "Flip Pending" condition. If a ne flip operation has completed (i.e., the new front buffer registers).



	MI_WAIT_FOR_EVEN	т
0	Reserved	
	Format:	MBZ



MI_WAIT_FOR_EVENT

Source:		Vide	oCS			
Length Bi	ias:	1				
The MI_W specific e Interface Note that this comr	VAIT_FC went oc in MI F t if a sp mand, t	DR_EVENT curs or wh unctions. C ecified cor he parser p	command is use ile a specific cor Only one event/c idition does not proceeds, treatin	ed to pause com ndition exists. Se condition can be exist (the condi ng this command	mand stream pr e Wait Events/C specified spe tion code is inac d as a no-operat	ocessing of this pipe only until a conditions, Device Programming ecifying multiple events is UNDEFINED. tive) at the time the parser executes ion.
DWord	Bit				Description	
0	31:29	Comman	d Type			
		Default V	alue:		0h MI_COMMA	ND
	28:23	MI Comm	nand Opcode			
		Default V	alue:	03h M	I_WAIT_FOR_EV	ENT
	22:20	Reserved				
		Format:				MBZ
	19:16	Condition	n Code Wait Sel	lect		
		This field These en	enables a wait f able select one o	for the duration of 15 condition of	that the corresp codes in the EXC	onding condition code is active. C register, that cause the parser to
		wait unti	that condition-	code in the EXC	C is cleared.	
		Value	Name		De	escription
		0h	Not enabled	Condition Code	e Wait Not Enab	led
		1h-5h	Enable	Condition Code	e select enabled;	; selects one of 5 codes, 0 - 4
		6h-15h	Reserved			
						
				Prog	gramming Note	25
		Note tha	t not all condition	on codes are imp n code is selecte	plemented. The p ad by this field T	barser operation is UNDEFINED if an The description of the EXCC register
		(Memory	Interface Regist	ters) lists the coo	les that are impl	lemented.
	15:0	Reserved				
		Format:				MBZ



		MI_V	VAIT_F	OR_E	VENT	
Source:		RenderCS				
Length B	ias:	1				
			Description			
pipe or Events/ can be Once p event/c is inacti comma If CSun MI_NOO MI_NOO MI_WA • B • N Events n (DE RRN flips or scan streame are required LOAD_R comma	IT_FOR ack-to- MI_WAIT_ conditional arsed, for conditional arsed, for arsed, for arsed, for ack-to- MI_WAIT_ must be MR 0x44 lines, pro- ar as the conditional conditional arsed, for arset for arse	il a specific event occurs o ions, Device Programming ed. Specifying multiple event the parser will halt (and su on occurs. Note that if a sp the time the parser execut a no-operation. iting for V-blank or flip do ting NOP register (or any of _EVENT under the following back MI_WAIT_FOR_EVENT of _FOR_EVENT is the last communasked in the Display Eng 4050) prior to waiting for the rior to starting the flip or load ey occur, so for improved power amming the DE RRMR regist R_IMMEDIATE	r while a spe g Interface in ents is UNDE spend comm pecified conc es this comm one, HW can other benign other benign other benign ommands mand before l gine Render R m with a MI_V ding the scanl wer savings it ter can be dor	ecific con <i>MI Func</i> FINED. nand arb lition doe nand, the go into F comman s: head = ta 	dition exists. See Wait tions. Only one event/condition witration) until the es not exist (the condition code e parser proceeds, treating this RC1/RC6 state. nd) must be set after il Mask Register R_EVENT command, or in the case of asked events will wake command mended to only unmask events that h MMIO or a	
DWord	Bit			Descript	ion	
0	31:29	Command Type		1		
		Default Value:		0h MI_C	OMMAND	
		Format:		OpCode		
	28:23	MI Command Opcode				
		Default Value:	03h M	II_WAIT_F	OR_EVENT	
		Format:	OpCo	de		
	22	Display Pipe C Horizontal	Blank Wait E	nable		
		Format:			Enable	
		This field enables a wait unt	til the start of	next Disp	lay Pipe C Horizontal Blank event occ	urs. This
		event is described as the sta cause a wait for up to a line.	rt of the next	Display C	Horizontal blank period. Note that th	nis can
	21	Display Pipe C Vertical Bla	nk Wait Enak	ole		
		Format:			Enable	



		MI_	WAIT_FOR_EVENT
	This field described for up to a	enables a wait u as the start of t an entire refresh	until the next Display Pipe C Vertical Blank event occurs. This event is he next Display C vertical blank period. Note that this can cause a wait period.
20	Display S	prite C Flip Per	nding Wait Enable
	Format:		Enable
	This field request is buffer add	enables a wait f pending, the pa dress has now b	for the duration of a Display Sprite C Flip Pending condition. If a flip arser will wait until the flip operation has completed (i.e., the new front een loaded into the active front buffer registers).
19:1	6 Conditio	n Code Wait Se	lect
	This field	enables a wait fo	or the duration that the corresponding condition code is active. These
	enable se	lect one of 15 co	ondition codes in the EXCC register, that cause the parser to wait until
	that cond	ition-code in the	e EXCC is cleared.
	value	Name	
		Not enabled	
	IN-5N	Enable	Condition Code Select Enabled; selects one of 5 codes, 0 - 4
	6n-15n	Reserved	
			Programming Notes
	Note tha unimpler (Memory	t not all condition nented condition Interface Regist	on codes are implemented. The parser operation is UNDEFINED if an n code is selected by this field. The description of the EXCC register ters) lists the codes that are implemented.
15	Display P	lane C Flip Pen	ding Wait Enable
	Format:		Enable
	This field request is buffer add	enables a wait f pending, the pa dress has now be	for the duration of a Display Plane C "Flip Pending" condition. If a flip arser will wait until the flip operation has completed (i.e., the new front een loaded into the active front buffer registers).
14	Display P	ipe C Scan Line	Wait Enable
	Format:	•	Enable
	This field defined as Compare	enables a wait v s the start of the Register.	while a Display Pipe C Scan Line condition exists. This condition is e scan line specified in the Pipe C Display Scan Line Count Range
13	Display P	ipe B Horizont	al Blank Wait Enable
-	Format:		Enable
	This field	enables a wait u	until the start of next Display Pipe B "Horizontal Blank" event occurs.
	This event can cause	t is described as a wait for up to	the start of the next Display B Horizontal blank period. Note that this a line.
12	Reserved		
	Format:		MBZ



11	Display Pipe B Vertical Blank Wait Enable	e
	Format:	Enable
	This field enables a wait until the next Disp described as the start of the next Display Pi wait for up to an entire refresh period.	lay Pipe B "Vertical Blank" event occurs. This ev pe B vertical blank period. Note that this can ca
10	Display Sprite B Flip Pending Wait Enabl	e
	Format:	Enable
	This field enables a wait for the duration of request is pending, the parser will wait unti buffer address has now been loaded into th	f a Display Sprite B "Flip Pending" condition. If a I the flip operation has completed (i.e., the new ne active front buffer registers).
9	Display Plane B Flip Pending Wait Enable	9
	Format:	Enable
	This field enables a wait for the duration of request is pending, the parser will wait unti buffer address has now been loaded into th	f a Display Plane B Flip Pending condition. If a f I the flip operation has completed (i.e., the new ne active front buffer registers.
8	Display Pipe B Scan Line Wait Enable	
	Format:	Enable
	This field enables a wait while a Display Pir	a B Scan Line condition exists. This condition is
	defined as the start of the scan line specifie Compare Register.	ed in the Pipe B Display Scan Line Count Range
7:6	defined as the start of the scan line specifie Compare Register.	ed in the Pipe B Display Scan Line Count Range
7:6	defined as the start of the scan line specifie Compare Register. Reserved Format:	MBZ
7:6	Anison and enables a wart while a Display Fig. defined as the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Enables	MBZ
7:6	Anison consists of what while d Display Fig. defined as the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Ena Format:	MBZ
7:6	Anison consists a wait while a Display rip defined as the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Ena Format: This field enables a wait until the start of meret is described as the start of the next D cause a wait for up to a line.	MBZ Benable Enable
7:6	Anison of the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Ena Format: This field enables a wait until the start of ne event is described as the start of the next D cause a wait for up to a line. Reserved	MBZ Able Enable Enable Enable A Horizontal Blank event occur Display A Horizontal blank period. Note that this
7:6	Initial chapters of what while distribution of property register defined as the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Ena Format: This field enables a wait until the start of mevent is described as the start of the next D cause a wait for up to a line. Reserved Format:	MBZ MBZ Enable ext Display Pipe A Horizontal Blank event occur Display A Horizontal blank period. Note that this MBZ
7:6 5 4	Anison of the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Ena Format: This field enables a wait until the start of mevent is described as the start of the next D cause a wait for up to a line. Reserved Format: Display Pipe A Vertical Blank Wait Enable	MBZ MBZ able Enable ext Display Pipe A Horizontal Blank event occur Display A Horizontal blank period. Note that this MBZ MBZ
7:6 5 4 3	Initial chapters of what while distribution of bisplay hip defined as the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Ena Format: This field enables a wait until the start of ne event is described as the start of the next D cause a wait for up to a line. Reserved Format: Display Pipe A Vertical Blank Wait Enable Format:	MBZ
7:6 5 4 3	Initial chapters of white white a Display Fip defined as the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Ena Format: This field enables a wait until the start of ne event is described as the start of the next D cause a wait for up to a line. Reserved Format: Display Pipe A Vertical Blank Wait Enable Format: Display Pipe A Vertical Blank Wait Enable Format: This field enables a wait until the next Display Pipe wait for up to an entire refresh period.	MBZ
7:6 5 4 3	Initial chapters of white white decision of the scan line specifie defined as the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Ena Format: This field enables a wait until the start of mevent is described as the start of the next D cause a wait for up to a line. Reserved Format: Display Pipe A Vertical Blank Wait Enable Format: Display Sprite A Flip Pending Wait Enable Display Sprite A Flip Pending Wait Enable	MBZ
7:6 5 4 3 2	Initial chapters of white white a Display Pipe defined as the start of the scan line specifie Compare Register. Reserved Format: Display Pipe A Horizontal Blank Wait Ena Format: This field enables a wait until the start of mevent is described as the start of the next D cause a wait for up to a line. Reserved Format: Display Pipe A Vertical Blank Wait Enable Format: Display Pipe A Vertical Blank Wait Enable Format: Display Pipe A Vertical Blank Wait Enable Format: Display Sprite A Flip Pending Wait Enable Format: Display Sprite A Flip Pending Wait Enable Format:	MBZ



MI_WAIT_FOR_EVENT								
request is pending, the parser will wait until the flip operation has completed (i.e., the ne buffer address has now been loaded into the active front buffer registers).								
1	Display Plane A 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).							
0	Display Pipe A Scan Line Wait Enable							
	Format:	Enable						
	This field enables a wait while a Display Pipe A "Scan Line" condition exists. This condition is defined as the start of the scan line specified in the Pipe A Display Scan Line Count Range Compare Register.							



mov - Move								
Source:	E	uIsa						
Length Bias	ength Bias: 4							
The mov ir types, forn channels c	The mov instruction moves the components in src0 into the channels of dst. If src0 and dst are of different types, format conversion is performed. If src0 is a scalar immediate, the immediate value is loaded into enabled channels of dst.							
A mov wit packed by raw move.	the same te destinatic	n region (B or L	JB type with	e, no source modif n HorzStride == 1	and ExecSize > 1) can only be written using			
Format: [(pred)] m	ov[.cmod] (e	exec_size) dst sr	c0					
			Program	nming Notes				
A <i>mov</i> inst destinatior	ruction with n value (in th	a source modif ne manner of a r	ier always c raw move).	copies a denorm so	purce value to a denorm			
There is no DWord int	o direct conv ermediate ty	ersion from B/L /pe.	JB to DF or	DF to B/UB. Use tv	vo instructions and a word or			
			Res	triction				
Restriction NaNs.	: Raw move	is not supporte	d for Float	values in ALT mode	e if any values are infinities or			
Restriction	: An accumu	lator can be a s	ource or de	estination operand	but not both.			
Restriction	: If the sour	e type and dest	tination typ	e differ, conditiona	al modifiers are not allowed.			
			S	yntax				
[(pred)]	mov[.cmod] (exec_size)	reg reg	[(pred)] mov[.c	mod] (exec_size) reg imm32			
				Pseudocode				
Evaluate src0.char	<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src0.chan[n]; } }</pre>							
Predicatio	on Conditio	nal Modifier S	Saturation	Source Modifier				
Y	Y	Y	(Y				
Src Types	Dst Types							
*B,*W,*D	*B,*W,*D							
*B,*W,*D	F							
F	*B,*W,*D							
F	F							
*W,*D	DF							
F	DF							
DF	*W,*D							
DF	F							



mov - Move							
DF D	F						
DWord	Bit			Description			
03	127:64	ImmSource					
		Exists If: ([Operand Controls][Src0.RegFile]=='IMM')					
		Format: EU_INSTRUCTION_SOURCES_IMM32					
	127:64	RegSource					
		Exists If: ([Operand Controls][Src0.RegFile]!='IMM')					
		Format:	EU_I	NSTRUCTION_SOURCES_REG			
	63:32	Operand Controls					
		Format:	at: EU_INSTRUCTION_OPERAND_CONTROLS				
	31:0	Header					
		Format:		EU_INSTRUCTION_HEADER			



movi - Move Indexed						
Source: EuIsa						
Length Bias: 4						
The movi instruction performs a fast component-wise indexed move for subfields from src0 to dst. The so operand must be an indirectly-addressed register. All channels of the source operand share the same register, which is provided by the register field of the first address subregister, with a possible immediate register offset. The register fields of the subsequent address subregisters are ignored by hardware. The subregister number of a source channel is provided by the subregister field of the corresponding address subregister, with a possible immediate subregister offset. The destination register may be either a directly-addressed or an indirectly-addressed register. This instruction effectively performs a subfield shuffling from one register to another. Up to eig subfields can be selected by an instruction.	ource jister s					
Format:						
Programming Notes						
The source register is calculated by adding the register portion of the first index register with the register portion of the address immediate, $a0.0[11:5] + addr_imm[9:5]$ For byte movi, byte0 of the destination is selected by $(a0.0[4:0])$, byte1 is selected by $(a0.1[4:0])$,, and byte7 is selected by $(a0.7[4:0])$. The rest of the bytes are undefined. For word movi, byte0 of the destination is selected by $(a0.0[4:1] & 0)$, byte1 is selected by $(a0.0[4:1] & 1)$, byte2 is selected by $(a0.1[4:1] & 0)$, byte3 is selected by $(a0.1[4:1] & 1)$,, and byte15 is selected by $(a0.7[4:1] & 1)$. The rest of the bytes are undefined. For DWord or float movi, byte0 of the destination is selected by $(a0.0[4:2] & 00b)$, byte1 is selected by $(a0.0[4:2] & 01b)$, byte2 is selected by $(a0.0[4:2] & 10b)$, byte3 is selected by $(a0.0[4:2] & 11b)$, byte4 is selected by $(a0.1[4:2] & 00b)$, byte5 is selected by $(a0.1[4:2] & 01b)$,, byte31 is selected by $(a0.7[4:2] & 11b)$. For all 3 conditions above, $a0 n[4:0] = a0 n[4:0] + addr imm[4:0]$						
Restriction						
Restriction: Source operand cannot be accumulators. The source operand must be a general register.						
Restriction: The source and destination must have the same type.						
Restriction: The execution size must be <= 8 (1, 2, 4, or 8).						
Restriction: The address register for the source must be aligned to the base (a0.0).						
Restriction: The destination register (directly or indirectly addressed) must be 16-byte aligned.						
Restriction: The destination region (directly or indirectly addressed) must point to the same GRF register.						
Restriction: The destination stride in bytes must equal the source element size in bytes.						
Restriction: The Align16 access mode is not allowed.						
Restriction: All the index registers (address subregisters) used must point to the same GRF register.						

Restriction: The instruction must use 1x1 indirect regioning.

Restriction: The destination offset is only used to create channel enables. Each element of the





movi - Move Indexed destination is directly mapped to the index registers for the movi instruction. i.e. a0.0 -> dst.0, a0.1 -> dst.1, a0.2 -> dst.2, etc. Restriction: Conditional Modifier is not allowed for this instruction. **Syntax** [(pred)] movi (exec_size) reg reg imm **Pseudocode** Evaluate(WrEn); srcregfile = regfile(src0); srcregbase = reg(address[0]) + reg(addr_imm); for (n = 0; n < RegWidth; n++) { if (WrEn.chan[n]) { srcsubreg =</pre> subreg(address[n] + addr_imm); dst.chan[n] = srcregfile.srcreg.srcsubreg; } } **Conditional Modifier Saturation Source Modifier** Predication Y Y Ν γ Src Types Dst Types В В UB UB W W UW UW D D UD UD F F **DWord** Bit Description 0..3 127:64 ImmSource Exists If: ([Operand Controls][Src0.RegFile]=='IMM') EU INSTRUCTION SOURCES IMM32 Format: 127:64 RegSource Exists If: ([Operand Controls][Src0.RegFile]!='IMM') Format: EU_INSTRUCTION_SOURCES_REG 63:32 **Operand Controls** Format: EU_INSTRUCTION_OPERAND_CONTROLS 31:0 Header

EU_INSTRUCTION_HEADER

Format:



mul - Multiply							
Source:	EuIsa						
Length Bias: 4							
		Des	escription				
The mul inst dst.	ruction performs compo	onent-wise m	nultiplication of src0 and src1 and stores the results in				
When both s The macro d multiplicatio	src0 and src1 are of type lescribed in the mach in n result.	e D or UD, on struction sho	nly the low 16 bits of each element of src1 are used. ould be used to obtain the full precision 64-bit				
Multiplicatio applicable flo	n of two floating-point oating-point mode.	numbers foll	llows the rules in mul - Multiply) based on the				
Format: [(pred)] mul	[.cmod] (exec_size) dst	src0 src1					
		Res	estriction				
Restriction: l	Jse a source modifier w	ith add to im	nplement subtraction.				
Restriction: V unsigned), th looks at the	When operating on inte ne destination cannot b low 34 bits of the result	gers with at le e floating-po).	least one of the source being a DWord type (signed or oint (implementation note: the data converter only				
Restriction: V unsigned), th (.sat) cannot	When operating on inte ne Overflow and Sign fla be used in this case.	gers with at lo ags are undef	least one source having a DWord type (signed or efined. Therefore, conditional modifiers and saturation				
Restriction: \	When multiplying a DW	and a W, the	e W has to be on src1, and the DW has to be on src0.				
			Syntax				
[(pred)] m	ul[.cmod] (exec_siz	e) reg reg	reg [(pred)] mul[.cmod] (exec_size) reg reg imm32				
	(pred), mar(.omod) (chec_prize) reg reg reg (pred), mar(.omod) (chec_prize) reg reg immoz						
			Pseudocode				
Evaluate(W src0.chan[rEn); for (n = 0; n] * srcl.chan[n];	n < exec_si } }	<pre>size; n++) { if (WrEn.chan[n]) { dst.chan[n] =</pre>				
Predication	Conditional Modifier	Saturation	n Source Modifier				
Υ	Υ	Y	Υ				
Src Types	Ost Types						
B B	3						
B V	N						
ВС)						
W V	N						
W C)						
W,D D)						
F F							



mul - Multiply								
DF D	۶							
DWord	Bit			Description				
03	127:64	ImmSource	1					
		Exists If: ([ImmSource][Src1.RegFile]=='IMM')						
		Format:	STRUCTION_SOURCES_REG_IMM					
	127:64	RegSource						
		Exists If: ([RegSource][Src1.RegFile]!='IMM')						
		Format:	EU_IN	STRUCTION_SOURCES_REG_REG				
	63:32	Operand Controls						
		Format: EU_INSTRUCTION_OPERAND_CONTROLS						
	31:0	Header						
		Format:	E	U_INSTRUCTION_HEADER				



	mac - Multiply Accumulate								
Source:	Eu	Isa							
Length Bias	: 4								
The mac in correspond	The mac instruction takes component-wise multiplication of src0 and src1, adds the results with the corresponding accumulator values, and then stores the final results in dst.								
Format: [(pred)] ma	Format: [(pred)] mac[.cmod] (exec_size) dst src0 src1								
				Restriction					
Restriction:	: Accumulato	r is an implic	it source and	l thus cannot be an	explicit source operand.				
				Syntax					
[(pred)]	<pre>mac[.cmod]</pre>	(exec_size	e) reg reg	reg [(pred)] ma	c[.cmod] (exec_size) reg reg imm32				
				Pseudocode					
Evaluate(src0.chan	WrEn); for [n] * src1	(n = 0; r .chan[n] +	n < exec_si acc0.chan[ze; n++) { if n]; } }	(WrEn.chan[n]) { dst.chan[n] =				
Predicatio	n Condition	nal Modifier	Saturation	Source Modifier					
Υ	Y		Y	Υ					
Src Types	Dst Types								
*B,*W	*B,*W,*D								
F	F								
DF	DF								
DWord	Bit			Des	cription				
03	127:64	ImmSou	rce						
		Exists If:	([Imm	nSource][Src1.RegF	ile]=='IMM')				
		Format:	EU_IN	ISTRUCTION_SOUF	RCES_REG_IMM				
127:64		RegSour	ce						
		Exists If:	([Reg	Source][Src1.RegFi	le]!='IMM')				
		Format:	EU_IN	NSTRUCTION_SOU	RCES_REG_REG				
	63:32	Operand	Controls						
		Format:	EU_INS	STRUCTION_OPERA	AND_CONTROLS				
	31.0	Header							

EU_INSTRUCTION_HEADER

Format:



mach - Multiply Accumulate High

Source: EuIsa Δ

Length Bias:

The mach instruction performs DWord integer multiply-accumulate operation and outputs the high DWord (bits 63:32).

For each enabled channel, this instruction multiplies the DWord in src1 with the high word of the DWord in src0, left shifts the result by 16 bits, adds it with the corresponding accumulator values, and keeps the whole 64-bit result in the accumulator. It then stores the high DWord (bits 63:32) of the results in dst.

This instruction is intended to be used to emulate 32-bit DWord integer multiplication by using the large number of bits available in the accumulator. For example, the following four instructions perform vector multiplication of two 32-bit signed integer sources from r2 and r3 and store the resulting vectors with the high 32 bits in r5 and the low 32 bits in r6.

mul (8) acc0:d r2.0<8;8,1>:d r3.0<8;8,1>:d //All channels must be enabled

mach (8) rTemp<1>:d r2.0<8;8,1>:d r3.0<8;8,1>:d //All channels must be enabled

mov (8) r5.0<1>:d rTemp<8;8,1>:d // High 32 bits mov (8) r6.0<1>:d acc0:d // Low 32 bits

The mul and mach instructions must have all channels enabled. The first mov should have channel enable from the destHI of IMUL, the second mov should have the channel enable from the destLO of IMUL.

As mach is used to generate part of the 64-bit DWord integer results, saturation modifier should not be used. In fact, saturation modifier should not be used for any of these four instructions.

Source and destination operands must be DWord integers. Source and destination must be of the same type, signed integer or unsigned integer.

If dst is UD, src0 and src1 may be UD and/or D. However, if any of src0 and src1 is D, source modifier (abs) must be present to convert it to match with dst.

If dst is D, src0 and src1 must also be D. They cannot be UD as it may cause unexpected overflow because the computed results are limited to 64 bits.

Format:

[(pred)] mach[.cmod] (exec_size) dst src0 src1

Restriction

Restriction: Accumulator is an implicit source and thus cannot be an explicit source operand.

Restriction: AccWrEn is required. The accumulator is an implicit destination and thus cannot be an explicit destination operand.

Syntax

[(pred)] mach[.cmod] (exec_size) reg reg reg [(pred)] mach[.cmod] (exec_size) reg reg imm32



mach - Multiply Accumulate High

Pseudocode

Eva	luate(WrEn);
-	

```
for ( n = 0; n < exec_size; n++ ) { acc.chan[n][63:0] = (src0.chan[n][31:16] *
src1.chan[n][31:0]) << 16 + acc.chan[n][63:0]; if ( WrEn.chan[n] ) { dst.chan[n][31:0] =
acc.chan[n][63:32]; } }</pre>
```

Predication Conditi		Condition	al Modifier	Saturatio	on Source N	lodifier	•	
Υ	Y N			Y	Y			
Src Types	Dst	t Types						
D	D							
UD	UD							
DWord		Bit				Des	scription	
03		127:64	ImmSou	rce				
			Exists If:	Exists If: ([Imn		[ImmSource][Src1.RegFile]=='IMM')		
			Format:	EU	J_INSTRUCTION_SOURCES_REG_IMM			
	127:64		RegSour	RegSource				
			Exists If:	([R	([RegSource][Src1.RegFile]!='IMM')		ile]!='IMM')	
			Format:	EU	EU_INSTRUCTION_SOURCES_REG_REG			
	63:32 Opera		Operand	Operand Controls				
		Format:	Format: EU_INS		JSTRUCTION_OPERAND_CONTROLS			
	31:0 Header		Header					
			Format:		EU_INSTRU	EU_INSTRUCTION_HEADER		



mad - Multiply Add

Source: EuIsa

Length Bias:

The mad instruction takes component-wise multiplication of src1 and src2, adds the results with the corresponding src0 values, and then stores the final results in dst.

Format:

[(pred)] mad[.cmod] (exec_size) dst src0 src1 src2

4

Restriction

Restriction: No explicit accumulator access because this is a three-source instruction. AccWrEn is allowed for implicitly updating the accumulator.

Restriction: All three-source instructions have certain restrictions, described in Instruction Machine Formats.

Syntax

[(pred)] mad[.cmod] (exec_size) reg reg reg reg

Pseudocode	
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src1.chan[n] * src2.chan[n] + src0.chan[n]; } }</pre>	

Predication 0		Cor	nditional Modifier		Saturation	Source Modifier				
Υ	Y			Y	Y					
Src Typ	oes C	Dst T	ypes							
F	F	:								
DF	۵	DF								
DWord	B	it				Descript	ion			
03	127:	:126	Rese	rved						
			Forn	nat:			MBZ			
	125:106 Source 2			ce 2						
			Forn	Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC						
	105 104:85		Rese	Reserved						
			Forn	nat:		MBZ				
			Sour	ce 1						
			Forn	nat: EU_I	NSTRUCTION	N_OPERAND_SRC_F	EG_THREE_SRC			
	8	4	Rese	rved						
	83:64		Forn	nat:	MBZ					
			Source 0							
			Forn	nat: EU_I	NSTRUCTION	N_OPERAND_SRC_F	EG_THREE_SRC			
	63:	:56	Desti	ination Regist	ter Number					
			Forn	nat:		DstRegNum	I			



		ma	ad - Multiply Add					
55:53	Destination Su	bregiste	r Number					
	Format:		DstSubRegNum[2:0]					
52:49	Destination Ch	Destination Channel Enable						
	Format:		ChanEn[4]					
	Four channel er	ables are	e defined for controlling which c	hannels are written into the				
	destination regi ExecSize channe bit is cleared, th enabled. Mnem where x corresp	Iestination region. These channel mask bits are applied in a modulo-four manner to all execSize channels. There is 1-bit Channel Enable for each channel within the group of 4. If the bit is cleared, the write for the corresponding channel is disabled. If the bit is set, the write is enabled. Mnemonics for the bit being set for the group of 4 are x, y, z, and w, respectively, where x corresponds to Channel 0 in the group and w corresponds to channel 3 in the group of a group of the						
48	Reserved							
	Format:	Format: MBZ						
47	NibCtrl			·				
	Format:		NibCtrl					
46	Reserved							
	Format: MBZ							
45:44	Destination Data Type							
	This field contains the data type for the destination							
	Value		Name					
	00b		Single Precision Float					
	01b	C	DWord					
	10b	l	Insigned DWord					
	11b	C	ouble Precision Float					
43:42	Source Data Ty	/pe	ata tupo for all three cources					
	Value			Jame				
	00b	S	Single Precision Float					
	01b		DWord					
	10b		Unsigned DWord					
	11b		Pouble Precision Float					
41.40	Source 2 Modi	fier						
12.10	Fxists If: ([Property[Source Modification]=='true')			e')				
	Format: SrcMod							
39.38	8 Source 1 Modifier							
55.50	Exists If: ([Property[Source Modification]=='true')							
	Format:	SrcMo	d					
41:36	Reserved	1						
	Exists If:	([Prope	erty[Source Modification]=='fals	e')				



mad - Multiply Add

	Format:	MBZ				
37:36	Source 0 Modifier					
	Exists If:	([Property[Source Modification]=='true')				
	Format:	SrcN	lod			
35	Reserved					
	Format:			MBZ		
34	Flag Register Number This field contains the flag register number for instructions with a non-zero Conditional Modifier.					
33	Flag Subregister Number This field contains the flag subregister number for instructions with a non-zero Conditional Modifier.					
32	Reserved					
	Format:			MBZ		
31:0	Header					
	Format:		EU_INSTRUCTION_HEADER			



nop - No Operation

Source: EuIsa

Length Bias:

Do nothing. The nop instruction takes an instruction dispatch but performs no operation. It can be used for assembly patching in memory, or to insert a delay in the program sequence.

Format:

nop

Restriction

Restriction: The nop instruction takes no instruction options other than Breakpoint.

nop

Syntax

Pseudocode

 $\{$; // The null statement, which does nothing. $\}$

Predication	Conditional Modifier	Saturation	Source Modifier
Ν	Ν	N	Ν

DWord	Bit	Description				
03	127:31	Reserved				
		Format: MBZ				
	30	DebugCtrl This field allows the insertion of a breakpoint at the current instruction. When the bit is set, hardware automatically stores the current IP in CR register and jumps to the System IP (SIP) BEFORE executing the current instruction.				
		Value		N	ame	
		0	No Breakpoint	[Default]		
		1	Breakpoint			
	29:7	Reserved				
		Format:	MBZ			
	6:0	Opcode				
		Format:		EU_OPCODE		



	PIPE_CONTROL								
Source:		Rend	erCS						
Length E	Bias:	2							
The PIPE	The PIPE_CONTROL command is used to effect the synchronization described above.								
DWord	Bit				C	Descripti	on		
0	31:29	Command Type							
		Default Va			3h GFXPIPE				
		Format:					Op	Code	
	28:27	Command	SubType						
		Default Va	lue:			3h G	FXPIF	PE_3D	
		Format:				OpCo	ode		
	26:24	3D Comma	and Opcode	9					
		Default Va	lue:		2	2h PIPE_C	ONT	[ROL	
		Format:			(OpCode			
	23:16	3D Comma	and Sub Op	code					
		Default Va	lue:		()h PIPE_C	CONT	FROL	
		Format:			(OpCode			
	15:8 Reserved								
		Format: MBZ						MBZ	
	7:0	DWord Length							
		Default Va	lue:		3h D	WORD_C	OUN	NT_n	
		Format: =n							
1	31:28	Reserved							
		Format:						МВД	
	27	Reserved							
Format:					МВД				
26 Reserved									
		Format:						MBZ	
	25	Reserved							
		Format:						МВД	
24 Destination Address Type Defines address space of Destination Address									
		Value	Name			Desc	ripti	on	
		0h	PPGTT	Use PPGTT add	dress	space for	DW	/ write	
		1h	GGTT	Use GGTT add	ress s	pace for	DW۱	write	
				1	Progr	amming	Not	tes	
		Ignored if	""No Write"	is selected in C	perat	tion.			



22 101	L PI Post Sync Operation							
	Value Name Description							
Oh	0h No LRI Operation No LRI operation occurs as a result of this instruction. The Post-Sync Operation field is valid and may be used to specify an operation							
1h	MMIO Write Immediate Data	Write the DWord contained in Immediate Data Low (DW3) to the MMIO offset specifed in the Address field.						
		Programming Notes						
This is se	bit caues a post sync at then the Post-Sync	operation with an LRI (Load Register Immediate) operation. If this Operation field must be cleared.						
2 Rese	erved							
1 Stor	e Data Index							
For	mat:	U1						
only	only applies to the Global HW status page. If this field is 1, the Destination Address Type in thi command must be set to 1 (GGTT).							
0 Com	Command Streamer Stall Enable							
For	Format: U1							
101	liat.	01						
If EN com com	IABLED, the sync oper pletion of those previo mand. This enables th	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command.						
If EN com com	IABLED, the sync oper pletion of those previo mand. This enables th	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes						
If EN com com	IABLED, the sync oper pletion of those previo mand. This enables th e of the following mus	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes It also be set:						
If EN com Com	IABLED, the sync oper- pletion of those previo mand. This enables th e of the following mus • Render Target Cac	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command.						
If EN com One	IABLED, the sync oper pletion of those previo mand. This enables th e of the following mus Render Target Cac Depth Cache Flush	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes at also be set: he Flush Enable ([12] of DW1) Enable ([0] of DW1) beard ((11 of DW1)						
If EN com One	IABLED, the sync oper pletion of those previo mand. This enables th e of the following mus • Render Target Cac • Depth Cache Flush • Stall at Pixel Scorel	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes at also be set: he Flush Enable ([12] of DW1) Enable ([0] of DW1) board ([1] of DW1) (EDW1)						
If EN com One	IABLED, the sync oper pletion of those previo mand. This enables th e of the following mus • Render Target Cac • Depth Cache Flush • Stall at Pixel Scorel • Depth Stall ([13] of • Post-Sync Operatio	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes at also be set: he Flush Enable ([12] of DW1) a Enable ([0] of DW1) board ([1] of DW1) f DW1) on ([13] of DW1)						
If EN com One	IABLED, the sync oper- pletion of those previo mand. This enables th e of the following mus • Render Target Cac • Depth Cache Flush • Stall at Pixel Scorel • Depth Stall ([13] of • Post-Sync Operatio	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes at also be set: the Flush Enable ([12] of DW1) a Enable ([0] of DW1) board ([1] of DW1) f DW1) on ([13] of DW1)						
If EN com One 9 Rese	IABLED, the sync oper pletion of those previo mand. This enables th e of the following mus Render Target Cac Depth Cache Flush Stall at Pixel Scorel Depth Stall ([13] of Post-Sync Operatio	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes at also be set: he Flush Enable ([12] of DW1) a Enable ([0] of DW1) board ([1] of DW1) f DW1) on ([13] of DW1)						
If EN com Com One 19 Rese	IABLED, the sync oper- pletion of those previo mand. This enables th e of the following mus • Render Target Cac • Depth Cache Flush • Stall at Pixel Scorel • Depth Stall ([13] of • Post-Sync Operations • Post-Sync Operations • Post-Sync Operations	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes It also be set: he Flush Enable ([12] of DW1) i Enable ([0] of DW1) board ([1] of DW1) f DW1) on ([13] of DW1)						
If EN com Com One L9 Rese L8 TLB For	IABLED, the sync oper pletion of those previo mand. This enables th e of the following mus • Render Target Cac • Depth Cache Flush • Stall at Pixel Scorel • Depth Stall ([13] of • Post-Sync Operations • Pos	ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes at also be set: he Flush Enable ([12] of DW1) a Enable ([0] of DW1) board ([1] of DW1) f DW1) on ([13] of DW1) U1						
9 Rese 8 TLB Forn If El com	IABLED, the sync oper- pletion of those previo mand. This enables th e of the following mus • Render Target Cac • Depth Cache Flush • Stall at Pixel Scorel • Depth Stall ([13] of • Post-Sync Operations • Po	U1 ation will not occur until all previous flush operations pending a ous flushes will complete, including the flush produced from this e command to act similar to the legacy MI_FLUSH command. Programming Notes It also be set: he Flush Enable ([12] of DW1) at Enable ([0] of DW1) board ([1] of DW1) f DW1) on ([13] of DW1) U1						



Programming Notes If ENABLED, all TLBs belonging to Render Engine will be invalidated once the flush operation is complete. Note that if the flush TLB invalidation mode is clear, a TLB invalidate will occur irrespective of this bit setting. 17 Reserved Format: MBZ 16 Generic Media State Clear Format: Disable If set, all generic media state context information will not be included with the next conte assuming no new state is initiated after the flush. If clear, the generic media state context state will not be affected. An MI_FLUSH with this bit set should be issued once all the Mee Objects that will be processed by a given persistent root thread have been issued or wher ML_SET_CONTEXT, once state is programmed, it will be saved and restarted as part context each time that context is saved/restored until an MI_FLUSH with this bit set is issu that context. 15:14 Post Sync Operation This field specifies an optional action to be taken upon completion of the synchronization operation. This field must be cleared if the LRI Post-Sync Operation bit is set. Value Name Description 0h No Write No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 115:14 Post Sync Operation 0h No Write the 64-bit PS_DEPTH_COUNT register to the Destination Address </th <th></th> <th>PIPE_CONTROL</th> <th></th> <th></th> <th></th>		PIPE_CONTROL							
If ENABLED, all TLBs belonging to Render Engine will be invalidated once the flush operation is complete. Note that if the flush TLB invalidation mode is clear, a TLB invalidate will occur irrespective of this bit setting. 17 Reserved Format: MBZ 16 Generic Media State Clear Format: Disable If set, all generic media state context information will not be included with the next conte assuming no new state is initiated after the flush. If clear, the generic media state context state will not be affected. An ML_FLUSH with this bit set should be issued once all the Med Objects that will be processed by a given persistent root thread have been issued or when ML_SET_CONTEXT switching from a generic media context to a 3D context completes. Wh using ML_SET_CONTEXT, once state is programmed, it will be saved and restarted as part context each time that context is saved/restored until an ML_FLUSH with this bit set is issue that context. 15:14 Post Sync Operation This field specifies an optional action to be taken upon completion of the synchronization operation. This field must be cleared if the LRI Post-Sync Operation bit is set. Value No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 1h Write Immediate Value No write the QWord containing Immediate Data Low, High DWs Data 1c He Destination Address 2h Write the 64-bit PS_DEPTH_COUNT register to the Count <th></th> <th>Programming Notes</th> <th></th> <th></th> <th></th>		Programming Notes							
17 Reserved Format: MBZ 16 Generic Media State Clear Format: Disable If set, all generic media state context information will not be included with the next conte assuming no new state is initiated after the flush. If clear, the generic media state context state will not be affected. An MI_FLUSH with this bit set should be issued once all the Mec Objects that will be processed by a given persistent root thread have been issued or when MI_SET_CONTEXT switching from a generic media context to a 3D context completes. Wh using MI_SET_CONTEXT, once state is programmed, it will be saved and restarted as part context each time that context is saved/restored until an MI_FLUSH with this bit set is issu that context. 15:14 Post Sync Operation 15:14 Post Sync Operation This field specifies an optional action to be taken upon completion of the synchronization operation. This field must be cleared if the LRI Post-Sync Operation bit is set. Value Name Oh No Write No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 1h Write the QWord containing Immediate Data Low, High DWs Data 2h Write the 64-bit PS_DEPTH_COUNT register to the Destination Address 3h Write Write the 64-bit TIMESTAMP register to the Destination Address 3h Write the 64-b	If ENABLED, all TLBs belonging to Render Engine will be invalidated once the flush operation is complete. Note that if the flush TLB invalidation mode is clear, a TLB invalidate will occur irrespective of this bit setting.								
Format: MBZ 16 Generic Media State Clear If set, all generic media state context information will not be included with the next conte assuming no new state is initiated after the flush. If clear, the generic media state context state will not be affected. An MI_FLUSH with this bit set should be issued once all the Mer Objects that will be processed by a given persistent root thread have been issued or when MI_SET_CONTEXT, once state is programmed, it will be saved and restarted as put on the state context is saved/restored until an MI_FLUSH with this bit set is issue that context. 15:14 Post Sync Operation 16 Using MI_SET_CONTEXT, once state is programmed, it will be saved and restarted as put on the synchronization operation. 17.15 This field specifies an optional action to be taken upon completion of the synchronization operation. 17.16 This field must be cleared if the LRI Post-Sync Operation bit is set. 18 Value Name 19 No Write No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 11 Write Immediate Write the QWord containing Immediate Data Low, High DWs to the Destination Address	<u></u>	Reserved							
If Generic Media State Clear Format: Disable If set, all generic media state context information will not be included with the next context state will not be affected. An MI_FLUSH with this bit set should be issued once all the Med Objects that will be processed by a given persistent root thread have been issued or when MI_SET_CONTEXT switching from a generic media context to a 3D context completes. Wh using MI_SET_CONTEXT, once state is programmed, it will be saved and restarted as part context each time that context is saved/restored until an MI_FLUSH with this bit set is issued to context. 5:14 Post Sync Operation This field specifies an optional action to be taken upon completion of the synchronization operation. This field specifies an optional action to be taken upon completion of the synchronization operation. This field must be cleared if the LRI Post-Sync Operation bit is set. Value Name Oh No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 1h Write Immediate Write the QWord containing Immediate Data Low, High DWs to the Destination Address 2h Write PS Depth Write the 64-bit PS_DEPTH_COUNT register to the Count 3h Write Write the 64-bit TIMESTAMP register to the Destination Address 3h Write Write the 64-bit TIMESTAMP register to the Destination Address		MBZ	at:	Format					
Format: Disable If set, all generic media state context information will not be included with the next context assuming no new state is initiated after the flush. If clear, the generic media state context state will not be affected. An MI_FLUSH with this bit set should be issued once all the Mer Objects that will be processed by a given persistent root thread have been issued or when MI_SET_CONTEXT switching from a generic media context to a 3D context completes. Wh using MI_SET_CONTEXT, once state is programmed, it will be saved and restarted as part context each time that context is saved/restored until an MI_FLUSH with this bit set is issued to context. 5:14 Post Sync Operation 5:14 Post Sync Operation This field specifies an optional action to be taken upon completion of the synchronization operation. This field must be cleared if the LRI Post-Sync Operation bit is set. Value Name Description 0h No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 1h Write Immediate Write the QWord containing Immediate Data Low, High DWs to the Destination Address 2h Write PS Depth Write the 64-bit PS_DEPTH_COUNT register to the Destination Address 3h Write Write the 64-bit TIMESTAMP register to the Destination Address 3h Write the 64-bit TIMESTAMP register to the Destination Address		ar	ric Media State Cle	Generi	16				
If set, all generic media state context information will not be included with the next context assuming no new state is initiated after the flush. If clear, the generic media state context state will not be affected. An MI_FLUSH with this bit set should be issued once all the Met Objects that will be processed by a given persistent root thread have been issued or when MI_SET_CONTEXT, switching from a generic media context to a 3D context completes. Wh using MI_SET_CONTEXT, once state is programmed, it will be saved and restarted as part context each time that context is saved/restored until an MI_FLUSH with this bit set is issued or the context. 5:14 Post Sync Operation This field specifies an optional action to be taken upon completion of the synchronization operation. This field must be cleared if the LRI Post-Sync Operation bit is set. Value Name 0h No Write No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 1h Write Immediate 2h Write PS Depth Write the 64-bit TS_DEPTH_COUNT register to the Count 2h Write the 64-bit TIMESTAMP register to the Destination Address 3h Write 1f Write the 64-bit TIMESTAMP register to the Destination		Disable	at:	Format					
Post Sync Operation Description This field specifies an optional action to be taken upon completion of the synchronization operation. This field must be cleared if the LRI Post-Sync Operation bit is set. Value Name Description Oh No Write No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 1h Write Immediate Write the QWord containing Immediate Data Low, High DWs to the Destination Address 2h Write PS Depth Write the 64-bit PS_DEPTH_COUNT register to the Count 3h Write Write the 64-bit TIMESTAMP register to the Destination Address 3h Write model address If executed in non-secure batch buffer, the address given will be in a PPGTT address space	edia en an 'hen t of any sued in	assuming no new state is initiated after the flush. If clear, the generic media state context save state will not be affected. An MI_FLUSH with this bit set should be issued once all the Media Objects that will be processed by a given persistent root thread have been issued or when an MI_SET_CONTEXT switching from a generic media context to a 3D context completes. When using MI_SET_CONTEXT, once state is programmed, it will be saved and restarted as part of any context each time that context is saved/restored until an MI_FLUSH with this bit set is issued in that context.							
Value Name Description 0h No Write No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 1h Write Immediate Write the QWord containing Immediate Data Low, High DWs to the Destination Address 2h Write PS Depth Write the 64-bit PS_DEPTH_COUNT register to the Destination Address 3h Write Write the 64-bit TIMESTAMP register to the Destination Address If executed in non-secure batch buffer, the address given will be in a PPGTT address space	Post Sync Operation Description This field specifies an optional action to be taken upon completion of the								
ValueNameDescriptionOhNo WriteNo write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc.1hWrite Immediate DataWrite the QWord containing Immediate Data Low, High DWs to the Destination Address2hWrite PS Depth CountWrite the 64-bit PS_DEPTH_COUNT register to the Destination Address3hWrite TimestampWrite the 64-bit TIMESTAMP register to the Destination AddressProgramming NotesIf executed in non-secure batch buffer, the address given will be in a PPGTT address space		synchronization operation. This field must be cleared if the LRI Post-Sync Operation bit is set.							
0h No Write No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc. 1h Write Immediate Write the QWord containing Immediate Data Low, High DWs to the Destination Address 2h Write PS Depth Count Write the 64-bit PS_DEPTH_COUNT register to the Destination Address 3h Write Write the 64-bit TIMESTAMP register to the Destination Address If executed in non-secure batch buffer, the address given will be in a PPGTT address space]	Description	e Name	Value					
1h Write Immediate Write the QWord containing Immediate Data Low, High DWs to the Destination Address 2h Write PS Depth Write the 64-bit PS_DEPTH_COUNT register to the Destination Address 3h Write Write the 64-bit TIMESTAMP register to the Destination Address Programming Notes If executed in non-secure batch buffer, the address given will be in a PPGTT address space	1	No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc.	No Write	0h					
2h Write PS Depth Count Write the 64-bit PS_DEPTH_COUNT register to the Destination Address 3h Write Timestamp Write the 64-bit TIMESTAMP register to the Destination Address Programming Notes If executed in non-secure batch buffer, the address given will be in a PPGTT address space		Write the QWord containing Immediate Data Low, High DWs to the Destination Address	1h Write Immediate Write the QWord containing Immediate Data Low, High DWs Data to the Destination Address						
3h Write Timestamp Write the 64-bit TIMESTAMP register to the Destination Address Programming Notes If executed in non-secure batch buffer, the address given will be in a PPGTT address space		Write the 64-bit PS_DEPTH_COUNT register to the Destination Address	2h Write PS Depth Write the 64-bit PS_DEPTH_COUNT register to the Count Destination Address						
Programming Notes If executed in non-secure batch buffer, the address given will be in a PPGTT address space		Write the 64-bit TIMESTAMP register to the Destination Address	3hWrite TimestampWrite the 64-bit TIMESTAMP register to the D Address						
If executed in non-secure batch buffer, the address given will be in a PPGTT address space		Programming Notes							
a secure ring or batch, address given will be in GGTT space									
13 Depth Stall Enable			h Stall Enable	Depth	13				
Format: Enable		Enable	at:	Format					
This bit should be set when obtaining a "visible pixel" count to preclude the possible inclu the PS_DEPTH_COUNT value written to memory of some fraction of pixels from objects in after the PIPE_CONTROL command	lusion in initiated	n obtaining a "visible pixel" count to preclude the possible incl ue written to memory of some fraction of pixels from objects i command	it should be set who S_DEPTH_COUNT va	This bit the PS_					
Value Name Description	1	Description	e Name	Value					



			PI	PE_CONTRO	OL			
	0h Disable 3D pipeline will not stall subsequent primitives at the Depth Test stage.							
	1h	Enable	3D pipeline wi until the Sync	ill stall any subseque and Post-Sync opera	nt primitives at the Depth Test stage ations complete.			
				Programmin	g Notes			
	This bit	t should	be DISABLED f	or operations other t	han writing PS_DEPTH_COUNT.			
	This bit comma	t will hav and issue	e no effect (be ed to the Media	sides preventing writ a pipe.	te cache flush) if set in a PIPE_CONTROI	Ĺ		
12	Render	Target	Cache Flush Ei	nable				
	Format				Enable			
	Setting point co from op this syn	Setting this bit will force Render Cache to be flushed to memory prior to this synchronization point completing. This bit should be set for all write fence sync operations to assure that results from operations initiated prior to this command are visible in memory once software observes this synchronization						
	Value	9	Name		Description			
	0h	Disat	ole Flush	Render Target Cach	ne is NOT flushed.			
	1h Enable Flush Render Target Cache is flushed.							
	Programming Notes							
	This bit should be DISABLED for End-of-pipe (Read) fences, PS_DEPTH_COUNT or TIMESTAMP queries.							
	This bit	is bit must not be set when Depth Stall Enable bit is set in this packet.						
11	Instruction Cache Invalidate Enable							
	Format				Enable			
	Setting this bit is independent of any other bit in this packet. This bit controls the invalidation of the L1 and L2 at the top of the pipe i.e. at the parsing time.							
10	Texture	e Cache	Invalidation E	nable	-			
	Format	:			Enable			
	Setting the text	Setting this bit is independent of any other bit in this packet. This bit controls the invalidation of the texture caches at the top of the pipe i.e. at the parsing time.						
9	Indirec	t State P	ointers Disab	le				
	Format				Enable			
	Description							
	At the completion of the post-sync operation associated with this pipe control packet, the indirect state pointers in the hardware are considered invalid; the indirect pointers are not saved in the context. If any new indirect state commands are executed in the command stream while the pipe control is pending, the new indirect state commands are preserved.							



			PIPE_CONTR	OL					
	Using Invalidate State Pointer (ISP) only inhibits context restoring of Push Constant (3DSTATE_CONSTANT_*) commands. Push Constant commands are only considered as Indirect State Pointers. Once ISP is issued in a context, SW must initialize by programming push constant commands for all the shaders (at least to zero length) before attempting any rendering operation for the same context.								
8	Notify Enable								
	Format:			Enable					
	If ENABLED Control reg Memory In	D, a Sync Con gisters) once Iterface Regis	npletion Interrupt will be ge the sync operation is comp sters for details.	enerated (if enabled by the MI Interrup lete. See Interrupt Control Registers ir	pt n				
7	Pipe Cont	rol Flush Ena	able						
	Format:			Enable					
	If ENABLEE from post	D, the PIPE_Co sync circles a	ONTROL command will wai re complete before execution	t until all previous writes of immediate ng the next command.	e data				
6	Reserved								
5	DC Flush I	nable							
	Format:	Format: Enable							
	Setting this bit enables flushing of the L3\$ portions that caches DC writes.								
4	VF Cache	Invalidation	Enable						
	Format:			Enable					
	Setting this VF address	s bit is indepo based cache	endent of any other bit in tl e at the top of the pipe i.e. a	his packet. This bit controls the invalid at the parsing time.	lation of				
3	Constant	Cache Invalio	dation Enable						
	Format:			Enable					
	Setting this bit is independent of any other bit in this packet. This bit controls the invalidation of the constant cache at the top of the pipe i.e. at the parsing time.								
2	State Cach	ne Invalidati	on Enable						
	Format:			Enable					
	Sotting thi	etting this bit is independent of any other bit in this packet. This bit controls the invalidation of ne L1 and L2 state caches at the top of the pipe i.e. at the parsing time.							
	the L1 and	L2 state cach	hes at the top of the pipe i.e	e. at the parsing time.					
1	the L1 and	L2 state cach	hes at the top of the pipe i.e	e. at the parsing time.					
1	Setting the the L1 and Stall At Pi	L2 state cach	hes at the top of the pipe i.e	e. at the parsing time.					
1	Setting this the L1 and Stall At Pi Format: Defines the	L2 state cach	hes at the top of the pipe i.e ard PIPE_CONTROL command	e. at the parsing time. Enable at the pixel scoreboard.					
1	Setting the the L1 and Stall At Pice Format: Defines the Value	L2 state cach xel Scoreboa e behavior of Name	hes at the top of the pipe i.e ard PIPE_CONTROL command	e. at the parsing time. Enable at the pixel scoreboard.					
1	Setting this the L1 and Stall At Pi Format: Defines the Value Oh	L2 state cach xel Scoreboa e behavior of Name Disable	hes at the top of the pipe i.e ard PIPE_CONTROL command De Stall at the pixel scoreboar	e. at the parsing time. Enable at the pixel scoreboard. escription rd is disabled.					



PIPE_CONTROL

				Programming Notes							
		This bit should be DISABLED for End-of-pipe (Read) fences, PS_DEPTH_COUNT or TIMESTAMP queries. This bit is ignored if Depth Stall Enable is set. Further the render cache is not flushed even if Write Cache Flush Enable bit is set.									
	0	Depth Cache Flush Enable									
		Format: Enable									
	Setting this bit enables flushing (i.e. writing back the dirty lines to memory ar tags) of depth related caches. This bit applies to HiZ cache. Stencil cache and										
		Value	Name	Description							
		0h	Flush Disabled	Depth relates caches (HiZ, Stencil and Depth) are NOT flushed.							
		1h	Flush Enabled	Depth relates caches (HiZ, Stencil and Depth) are flushed.							
				Programming Notes							
		Ideally depth caches need to be flushed only when depth is required to be coherent in memory for later use as a texture, source or honoring CPU lock. This bit should be DISABLED for End-of-pipe (Read) fences, PS_DEPTH_COUNT or TIMESTAMP queries.									
2	31:2	Addres	S								
		GraphicsAddress[31:2]U32									
		QW address of where the Immediate Data following this DW in the packet to be stored. Bit 2 MBZ Ignored if "No Write" is the selected in Post-Sync Operation If LRI Post-Sync Operation is set: Bits 22:2 (Bits 31:23 are reserved MBZ) specify the MMIO offset destination for the data in the Immediate Data Low (DW3) field. Only DW writes are valid.									
	1:0	Reserved									
		Format	:	MBZ							
3	31:0	Immed	iate Data								
		Format	:	U32							
		This fiel Post-Sy Ignored	This field specifies the Lower DWord value to be written to the targeted location. Only valid when Post-Sync Operation is 1h (Write Immediate Data) or LRI Post-Sync Operation is set. gnored if Post-Sync Operation is "No write", "Write PS_DEPTH_COUNT" or "Write TIMESTAMP".								
		This field should be programmed to 0 when Post-Sync Operation is set to Write PS Depth Count or Write Timestamp.									
4	31:0	Immed	iate Data								
		Format		U32							
		This fiel when Po write", "	d specifies the L ost-Sync Operat Write PS_DEPTH	Jpper DWord value to be written to the targeted location. Only valid tion is 1h (Write Immediate Data) Ignored if Post-Sync Operation is "No H_COUNT", "Write TIMESTAMP" or "LRI Post Sync Opeation".							
				Programming Notes							
		This fie	ld should be pro	ogrammed to 0 when Post-Sync Operation is set to Write PS							



PIPE_CONTROL

Depth Count or Write Timestamp.



PIPELINE_SELECT								
Source:		В	Spec					
Length B	Bias:	1						
Description								
The PIPELINE_SELECT command is used to specify which GPE pipeline is to be considered the 'current' active pipeline. Issuing 3D-pipeline-specific commands when the Media pipeline is selected, or vice versa, is UNDEFINED.								
Issuing 3D-pipeline-specific commands when the GPGPU pipeline is selected, or vice versa, is UNDEFINED.								
Programming common non pipeline commands (e.g., STATE_BASE_ADDRESS) is allowed in all pipeline modes.								
				Programming Notes				
Software must ensure all the write caches are flushed through a stalling PIPE_CONTROL command followed by another PIPE_CONTROL command to invalidate read only caches prior to programming MI_PIPELINE_SELECT command to change the Pipeline Select Mode. Example:								
Workload-3Dmode PIPE_CONTROL (CS Stall, Depth Cache Flush Enable, Render Target Cache Flush Enable, DC Flush Enable) PIPE_CONTROL (Constant Cache Invalidate, Texture Cache Invalidate, Instruction Cache Invalidate, State Cache invalidate)								
PIPELIN	IE_SELE	CI (GP	GPU)					
Dword	BIT	C		Descrip	tion			
0	31:29	Default	ana iyp t Value:	e	3h GEXPIPE			
	20.27	Commu	and Cub	Trum a				
	28.27	Default Value: 1h GFXPIPE SINGLE DW						
	26:24	3D Con	nmand	Dpcode				
		Valu	ue	Name				
		1h	G	EXPIPE_NONPIPELINED [Default]				
	23:16	3D Con	nmand	Sub Opcode				
		Defaul	t Value:		04h GFXPIPE			
	15:2	Reserve	ed					
	1:0	Pipelin	e Select					
		Value	Name	Des	cription			
		0	3D	3D pipeline is selected				
		1	Media	Media pipeline is selected (Includ video playback, and generic medi	es HD optical disc playback, HD a workloads)			
		2	GPGPU	GPGPU pipeline is selected				





pln - Plane

Source: EuIsa

Length Bias:

The pln instruction computes a component-wise plane equation ($w = p^*u + q^*v + r$ where u/v/w are vectors and p/q/r are scalars) of src0 and src1 and stores the results in dst. src1 is the input vector u.

src0 provides input scalars p, q, and r, where p is the scalar value based on the region description of src0 and q and r are the scalar values implied from the src0 region. Specifically, g is the second component and r is the fourth component of the 4-tuple (128-bit aligned) that p belongs to.

Format:

[(pred)] pln[.cmod] (exec_size) dst src0 src1

4

Restriction

Restriction: This is a specialized instruction that only supports an execution size (ExecSize) of 8 or 16.

Restriction: The src0 region must be a replicated scalar (with HorzStride == VertStride == 0).

Restriction: src0 must specify .0 or .4 as the subregister number, corresponding to a subregister byte offset of 0 or 16.

Restriction: Source operands cannot be accumulators.

Syntax

[(pred)] pln[.cmod] (exec_size) reg reg reg

Pseudocode

```
Evaluate(WrEn);
     for ( n = 0; n < exec_size; n++ ) {</pre>
         float dwP = src0.RegNum.SubRegNum[bits4:2];
                                                              // A DWord-aligned scalar.
         float dwQ = src0.RegNum.(SubRegNum[bit4:2] | 0x1); // Second component.
         float dwR = src0.RegNum.(SubRegNum[bit4:2] | 0x3); // Fourth component.
         if ( ExecSize == 8 ) {
             u = src1.RegNum
             v = srcl.(RegNum + 1)
         } else {
             if (n < 8) {
                 u = srcl.RegNum
                 v = src1.(RegNum + 1)
             } else {
                 u = src1.(RegNum + 2)
                 v = src1.(RegNum + 3)
             }
         }
         if ( WrEn.chan[n] ) {
            dst.chan[n] = dwP * u.chan[n] + dwQ * v.chan[n] + dwR;
         }
     }
Predication
           Conditional Modifier Saturation Source Modifier
           γ
                               Y
                                         Ν
Src Types Dst Types
```

v



pin - Plane							
F F	-						
DWord	Bit	Description					
03	127:64	ImmSource					
		Exists If:	mSource][Src1.RegFile]=='IMM')				
		Format: EU_INSTRUCTION_SOURCES_REG_IMM					
	127:64		RegSource				
			Exists If: ([RegSource][Src1.RegFile]!='IMM')				
		Format:	Format: EU_INSTRUCTION_SOURCES_REG_REG				
63:32 Operand Controls							
Format: EU_INSTRUCTION_OPERAND_CON			STRUCTION_OPERAND_CONTROLS				
	31:0	Header					
		Format:		EU_INSTRUCTION_HEADER			


			re	t - Return				
Source:	EuIsa	a						
Length Bias:	4							
			Des	scription				
Return execu	ution to the co	ode sequen	ce that calle	d a subroutine.				
The ret instr return IP in t predicated, t correspondir instruction, t	The ret instruction can be predicated or non-predicated. If non-predicated, all channels jump to the return IP in the first channel of src0 and restore CallMask from the second channel of src0. If predicated, the enabled channels jump to the return IP from the first channel of src0 and the corresponding bits in the CallMask are cleared to zero; if all CallMask bits are zero after the ret instruction, then execution jumps to the return IP from the first channel of src0.							
When SPF is	s on, the pred	ication cont	rol must be	scalar.				
Format: [(pred)] ret ((exec_size) nu	ll src0						
				Restriction				
Restriction: 1	This instructio	n cannot tal	ke accumula	ator as source.				
Restriction: 1	The src0 regio	oning contro	l must be <	2;2,1>,				
				Constant				
[(prod)] r	ot (orog gi			Syntax				
[(pred)] I	et (exec_si	.ze) null i	leg					
			Pse	udocode				
Evaluate(W src0.chan[n < 32; n+ are zero J	rEn); for (0]; CallMas +) { PcIP[ump(src0.ch	n = 0; n k[n] = 0; n] = IP + aan[0]); Ca	< exec_si } else { 1; } if (allMask =	<pre>ze; n++) { if PcIP[n] = IP + CallMask[n:0] src0.chan[1]; }</pre>	<pre>(WrEn.chan[n]) { PcIP[n] = 1; } } for (n = exec_size; == 0)) { // all channels</pre>			
Predication	Conditional	Modifier	Saturation	Source Modifier				
Υ	Ν		N	Ν				
Src Types D,UD								
DWord	Bit	Description						
03	127:64	ImmSourc	e					
		Exists If:	([Oper	and Controls][Src0.	RegFile]=='IMM')			
		Format:	EU_INS	STRUCTION_SOURC	CES_IMM32			
	127:64	RegSource	2					
		Exists If:	([Oper	and Controls][Src0.	RegFile]!='IMM')			
		Format:	EU_IN	STRUCTION_SOUR	CES_REG			
	63:32	Operand C	Controls					
		Format:	EU_INS	STRUCTION_OPERA	AND_CONTROLS			
	31:0	Header						



ret - Return						
		Format:	EU_INSTRUCTION_HEADER			



			rndd	-	Round De	own	
Source:	EuIs	а					
Length Bias:	4						
The rndd instruction takes component-wise floating point downward rounding (to the integral float number closer to negative infinity) of src0 and storing the rounded integral float results in dst. This is commonly referred to as the floor() function. Each result follows the rules in the following tables based on the floating-point mode.							
Format: [(pred)] rndc	l[.cmod] (exe	ec size) dst s	src0				
		_ ,	Rec	tri	iction		
Restriction: N	lo accumulat	or access, i	mplicit or	ex	xplicit.		
					Syntax		
[(pred)] rr	ndd[.cmod]	(exec_siz	e) reg	reg	g [(pred)] rndd[.cmod] (exec_size) reg i	.mm32
					Pseudocode		
Evaluate(Wr floor(src0.	cEn); for (.chan[n]);	n = 0; n } }	< exec	_si	ize; n++) { if	(WrEn.chan[n]) { dst.c	han[n] =
Predication	Conditiona	Modifier	Saturati	on	Source Modifier		
Y	Y		Y		Υ		
Src Types D	st Types						
F F							
DWord	Bit				Des	cription	
03	127:64	ImmSour	ce				
		Exists If:	([O	ber	rand Controls][Src0.	RegFile]=='IMM')	
		Format:	EU_	IN:	STRUCTION_SOUR	CES_IMM32	
	127:64	RegSourc	e				
		Exists If:	([O	([Operand Controls][Src0.RegFile]!='IMM')			
		Format:	EU	IN	ISTRUCTION_SOUR	CES_REG	
	63:32	Operand	Controls				
		Format:	EU_	IN:	STRUCTION_OPERA	AND_CONTROLS	
	31:0	Header		-			
		Format:		E	EU_INSTRUCTION_H	HEADER	



rnde - Round to Nearest or Even

Source: EuIsa

Length Bias:

The rnde instruction takes component-wise floating point round-to-even operation of src0 with results in two pieces - a downward rounded integral float results stored in dst and the round-to-even increments stored in the rounding increment bits. The round-to-even increment must be added to the results in dst to create the final round-to-even values to emulate the round-to-even operation, commonly known as the round() function. The final results are the one of the two integral float values that is nearer to the input values. If the neither possibility is nearer, the even alternative is chosen.

Each result follows the rules in the following tables based on the floating-point mode.

Format:

[(pred)] rnde[.cmod] (exec_size) dst src0

Restriction

Restriction: No accumulator access, implicit or explicit.

Syntax

[(pred)] rnde[.cmod] (exec_size) reg reg [(pred)] rnde[.cmod] (exec_size) reg imm32

Y

Pseudocode

```
Evaluate(WrEn); for ( n = 0; n < exec_size; n++ ) { if ( WrEn.chan[n] ) { if (
    src0.chan[n] - floor(src0.chan[n]) > 0.5f ) { dst.chan[n] = floor(src0.chan[n]) + 1; }
else if ( src0.chan[n] - floor(src0.chan[n]) < 0.5f ) { dst.chan[n] = floor(src0.chan[n]);
} else { if ( floor(src0.chan[n]) is odd ) { dst.chan[n] = floor(src0.chan[n]) + 1; } else
{ dst.chan[n] = floor(src0.chan[n]); } } }</pre>
```

Predication Conditional Modifier Saturation Source Modifier

Y

Src Types Dst Types

γ

F

Y

DWord	Bit		Description		
03	127:64	ImmSource			
		Exists If:	([Operand Controls][Src0.RegFile]=='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_IMM32		
	127:64	RegSource			
		Exists If:	([Operand Controls][Src0.RegFile]!='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_REG		
	63:32	Operand Controls			
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
	31:0	Header			



rnde - Round to Nearest or Even						
	Format:	EU_INSTRUCTION_HEADER				



rndz -	Round	to Zero

Source: EuIsa

Length Bias:

The rndz instruction takes component-wise floating point round-to-zero operation of src0 with results in two pieces - a downward rounded integral float results stored in dst and the round-to-zero increments stored in the rounding increment bits. The round-to-zero increment must be added to the results in dst to create the final round-to-zero values to emulate the round-to-zero operation, commonly known as the truncate() function. The final results are the one of the two closest integral float values to the input values that is nearer to zero.

Format:

[(pred)] rndz[.cmod] (exec_size) dst src0

Restriction

Restriction: No accumulator access, implicit or explicit.

Syntax

[(pred)] rndz[.cmod] (exec_size) reg reg [(pred)] rndz[.cmod] (exec_size) reg imm32

Pseudocode								
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = floor(src0.chan[n]); if (abs(src0.chan[n]) < abs(dst.chan[n])) { dst.chan[n] = floor(src0.chan[n]) + 1; } else { dst.chan[n] = floor(src0.chan[n]); } }</pre>								
Predication	Conditional Modifier	Saturation	Source Modifier					
Y	Υ	Υ	Υ					

Src Types Dst Types

F	F				
DWord	Bit		Description		
03	127:64	ImmSource			
		Exists If:	([Operand Controls][Src0.RegFile]=='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_IMM32		
	127:64	RegSource	RegSource		
		Exists If:	([Operand Controls][Src0.RegFile]!='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_REG		
	63:32	Operand Co	Operand Controls		
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
	31:0	Header	Header		
		Format:	EU_INSTRUCTION_HEADER		



			rndu	- Round l	Jp	
Source:	EuIsa	a				
Length Bias:	4					
The rndu inst to positive in Each result f	ruction takes finity) of src0 ollows the ru	componer , commonly les in the fc	nt-wise floati y known as t bllowing tabl	ing point upward ro he ceiling() functio les based on the flo	ounding (to the integral float n. pating-point mode.	number closer
Format: [(pred)] rndu	ı[.cmod] (exe	c_size) dst s	src0			
			Restric	ction		
Restriction: N	lo accumulat	or access, ii	mplicit or ex	plicit.		
				Syntax		
[(pred)] rr	ndu[.cmod]	(exec_siz	e) reg reg	[(pred)] rndu[.cmod] (exec_size) reg i	.mm32
				Pseudocode		
Evaluate(Wn src0.chan[r else { dst.	rEn); for (n] - floor(.chan[n] =	n = 0; n src0.chan src0.chan	< exec_si [n]) > 0.0 [n]; } } }	ze; n++) { if f) { dst.chan[(WrEn.chan[n]) { if (n] = floor(src0.chan[n])	+ 1; }
Predication	Conditiona	Modifier	Saturation	Source Modifier		
Y	Y		Y	Y		
Src Types D	st Types					
F F						
DWord	Bit			Des	cription	
03	127:64	ImmSour	ce			
		Exists If:	([Opera	and Controls][Src0.	RegFile]=='IMM')	
		Format:	EU_INS	STRUCTION_SOURC	CES_IMM32	
	127:64	RegSourc	e			
		Exists If:	([Oper	and Controls][Src0.	RegFile]!='IMM')	
		Format:	EU_INS		CES_REG	
	63:32	Operand	Controls			
		Format:	EU_INS	STRUCTION_OPERA	ND_CONTROLS	
	31:0	Header				
		Format:	E	U_INSTRUCTION_F	IEADER	



sel - Select
Source: EuIsa
Length Bias: 4
Description
The sel instruction selectively moves the components in src0 or src1 into the channels of dst based on the predication. On a channel by channel basis, if the channel condition is true, data in src0 is moved into dst. Otherwise, data in src1 is moved into dst.
As the predication is used to select the two sources, it is not included in the evaluation of WrEn. The predicate clause is mandatory if cmod is omitted/0000b. If both predication and the conditional modifier are omitted, the results are undefined.
If the conditional modifier is specified (not 0000b, a compare is performed and the resulting condition flag is used for the sel instruction. Conditional modifiers .ge and .l follow the cmpn rules, and all other conditional modifiers follow the cmp rules. Predication is not allowed in this mode.
A sel instruction with cmod .l is used to emulate a MIN instruction.
A sel instruction with cmod .ge is used to emulate a MAX instruction.
For a sel instruction with a .l or .ge conditional modifier, if one source is NaN and the other not NaN, the non-NaN source is the result. If both sources are NaNs, the result is NaN. For all other conditional modifiers, if either source is NaN then src1 is selected.
A sel instruction without a conditional modifier always copies a denorm source value to a denorm destination value (in the manner of a raw move).
The sel instruction uses any conditional modifier internally and does not update the flag register if a conditional modifier is used.
A sel instruction with a conditional modifier flushes any selected denorm source value to a zero destination value.
Format: (pred) sel[.cmod] (exec_size) dst src0 src1
Restriction
Restriction: The maximum execution size is 16. SIMD32 is not supported.
Syntax
(pred) sel[.cmod] (exec_size) reg reg reg (pred) sel[.cmod] (exec_size) reg reg imm32
Pseudocode
<pre>Evaluate(WrEn, NoPMask); if (cmod == "0000") { // no CMod Evaluate(PMask); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { if (PMask.channel[n]) { dst.chan[n] = src0.chan[n]; } else { dst.chan[n] = src1.chan[n]; } } } } else { // with CMod Evaluate(CMod); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { if (CMod.chan[n]) { dst.chan[n] = src0.chan[n]; } else { dst.chan[n] = src1.chan[n]; } } } }</pre>



sel - Select							
Predicatio	n Co	onditional	Modifier	Saturation	n Source Modifier		
Υ	Y			Υ	Υ		
Src Types	Dst	Types					
*B,*W*D	*B,*\	<i>N</i> ,*D					
F	F						
DF	DF						
DWord		Bit	Description				
03		127:64	ImmSource				
			Exists If:	([Im	([ImmSource][Src1.RegFile]=='IMM')		
			Format: EU_INSTRUCTION_SOURCES_REG_IMM				
		127:64	RegSource				
			Exists If:	([Re	([RegSource][Src1.RegFile]!='IMM')		
			Format:	EU_I	INSTRUCTION_SOURCES_REG_REG		
		63:32	Operand Controls				
			Format:	EU_INSTRUCTION_OPERAND_CONTROLS			
		31:0	Header				
			Format:		EU_INSTRUCTION_HEADER		



send - Send Message

Source: EuIsa Length Bias: 4

Description

Send a message stored in GRF starting at <src> to a shared function identified by <ex_desc> along with control from <desc> with a GRF writeback location at <dest>.

The send instruction performs data communication between a thread and external function units, including shared functions (Sampler, Data Port Read, Data Port Write, URB, and Message Gateway) and some fixed functions (e.g. Thread Spawner, who also have an unique Shared Function ID). The send instruction adds an entry to the EU's message request queue. The request message is stored in a block of contiguous GRF registers. The response message, if present, will be returned to a block of contiguous GRF registers. The return GRF writes may be in any order depending on the external function units. <src> is the lead GRF register for request. <dest> is the lead GRF register for response. The message descriptor field <desc> contains the Message Length (the number of consecutive GRF registers). It also contains the header present bit, and the function control signals. The extend mesage descriptor field <ex_desc> contains the target function ID. WrEn is forwarded to the target function in the message sideband.

The send instruction is the only way to terminate a thread. When the EOT (End of Thread) bit of <ex_desc> is set, it indicates the end of thread to the EU, the Thread Dispatcher and, in most cases, the parent fixed function.

Message descriptor field <desc> can be a 32-bit immediate, imm32, or a 32-bit scalar register, <reg32a>. GEN restricts that the 32-bit scalar register <reg32a> must be the leading dword of the address register. It should be in the form of a0.0<0;1,0>:ud. When <desc> is a register operand, only the lower 29 bits of <reg32a> are used.

<ex_desc> is a 6-bit immediate, imm6. The lower 4bits of the <ex_desc> specifies the SFID for the message. The MSb of the message descriptor, the EOT field, always comes from bit 127 of the instruction word, which is the MSb of imm6. A thread must terminate with a send instruction with EOT turned on.

<src> is a 256-bit aligned GRF register. It serves as the leading GRF register of the request.

<dest> serves for two purposes: to provide the leading GRF register location for the response message if present, and to provide parameters to form the channel enable sideband signals. <dest> signals whether there is a response to the message request. It can be either a null register, a direct-addressed GRF register or a register-indirect GRF register. Otherwise, hardware behavior is undefined.

If <dest> is null, there is no response to the request. Meanwhile, the Response Length field in <desc> must be 0. Certain types of message requests, such as memory write (store) through the Data Port, do not want response data from the function unit. If so, the posted destination operand can be null. If <dest> is a GRF register, the register number is forwarded to the shared function. In this case, the target function unit must send one or more response message phases back to the requesting thread. The number of response message phases must match the Response Length field in <desc>, which of course cannot be zero. For some cases, it could be an empty return message. An empty return message is defined as a single phase message with all channel enables turned off.

The subregister number, horizontal stride, destination mask and type fields of <dest> are always valid and are used in part to generate on the WrEn. This is true even if <dest> is a null register (this is an exception for null as for most cases these fields are ignored by hardware).

The 16-bit channel enables of the message sideband are formed based on the WrEn. Interpretation of



send - Send Message

the channel enable sideband signals is subject to the target external function. In general for a 'send' instruction with return messages, they are used as the destination dword write mask for the GRF registers starting at <dest>. For a message that has multiple return phases, the same set of channel enable signals applies to all the return phases.

Thread managed memory coherency: A special usage of using non-null <dest> is to support writecommit signaling for memory write service by the Data Port Write unit. If <post_dest> is not null for a memory write request, the Data Port along with the Data Cache or Render Cache will wait until all the posted writes for the request have reached the coherent domain before sending back to the requesting thread an empty message to <dest> register. A memory write reaching the coherent domain, also referred to as reaching the global observable state, means that subsequent read to the same memory location, no matter which thread issues the read, must return the data of the write. The destination dependency control, {NoDDClr}, can be used in this instruction. This allows software to control the destination dependencies for multiple 'read'-type messages similar to that for multiple instructions using EU execution pipeline. As send does not check register dependencies for the post destination, {NoDDChk} should not be used for this instruction.

Restriction

Restriction: Software must obey the following rules in signaling the end of thread using the send instruction:

The posted destination operand must be null.

No acknowledgement is allowed for the send instruction that signifies the end of thread. This is to avoid deadlock as the EU is expecting to free up the terminated thread's resource.

A thread must terminate with a send instruction with message to a shared function on the output message bus; therefore, it cannot terminate with a send instruction with message to the following shared functions: Sampler unit, NULL function

For example, a thread may terminate with a URB write message or a render cache write message. A root thread originated from the media (generic) pipeline must terminate with a send instruction with message to the Thread Spawner unit. A child thread should also terminate with a send to TS. Please refer to the Media Chapter for more detailed description.

The send instruction can not update accumulator registers.

Saturate is not supported for send instruction.

ThreadCtrl are not supported for send instruction.

The send with EOT should use register space R112-R127 for <src>. This is to enable loading of a new</src>	
hread into the same slot while the message with EOT for current thread is pending dispatch	

Predicati	ion	Cond	Conditional Modifier		Saturatio	n Sourc	e Modifier]	
Υ	Ν				Ν	Ν			
DWord	B	it					Descrip	tion	
03	127	7:96	Message						
			Format:		EU_INSTRUCTION_OPERAND_SEND_MSG				
	95	:89	Flags						
			Format:		EU_	NSTRUC	TION_FLAG	S	
	88	:64	Source 0						
			Exists If:	(Struc	cture[EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align1')				
			Format:	EU_IN	ISTRUCTIO	N_OPER/	AND_SRC_R	EG_ALIGN1	



		send - Sei	nd Message	e				
88:64	Source 0							
	Exists If: (Str	ucture[EU_INSTRUC	FION_CONTROLS_A][AccessMode]=='Align16')				
	Format: EU_	INSTRUCTION_OPER	RAND_SRC_REG_ALIG	N16				
63:32	Operand Cont	Dperand Control						
	Format:	EU_INSTRUCTION	LS					
31:28	Controls B	Controls B						
	Format:	EU_INSTRUCTION_CONTROLS_B						
27:24	Shared Function	on ID (SFID)						
	Format:			SFID				
23:8	Controls A							
	Format:	EU_INSTRUCTI	ON_CONTROLS_A					
7	Reserved	Reserved						
	Format:	MBZ						
6:0	Opcode		_					
	Format:		EU_OPCODE					



		:	shl	- Shift Lef	t		
Source:	EuIsa	1					
Length Bias:	4						
			Des	cription			
Perform com the results in	ponent-wise dst, inserting	logical left shift o zero bits in the	of the l numbe	bits in src0 by the sl er of LSBs indicated	hift count indicated in src1, storing by the shift count.		
Hardware de long as the sl	etects overflo nifted result i	w properly and u s within 33 bits. (ses it t Otherw	to perform any satu vise, the result is un	ration operation on the result, as defined.		
Note: For wo	ord and DWo	rd operands, the	accum	nulators have 33 bit	S.		
The shift cour unsigned inte	nt is taken fro eger in the ra	om the low five b nge 0 to 31.	its of s	src1, regardless of t	he src1 type and treated as an		
Format: [(pred)] shl[.c	cmod] (exec_s	size) dst src0 src1					
			Rest	riction			
Restriction: A	ccumulator c	annot be destina	tion, iı	mplicit or explicit.			
Restriction: R	esults of satu	iration in packed	-DWor	rd mode are unpred	licable.		
				Syntax			
[(pred)] sh	nl[.cmod] (exec size) req	req	req [(pred)] shi	l[.cmod] (exec size) req req :	imm32	
		_ , ,					
			Pseu	udocode			
Evaluate(Wr = srcl.char src0.chan[r	rEn); for (n[n] & 0x1F n] << shift	n = 0; n < ex ; // Always us Cnt; } }	ec_si e low	ze; n++) { if 7 5 bits for shift	(WrEn.chan[n]) {		
Predication	Conditional	Modifier Satur	ation	Source Modifier			
Υ	γ	Y		Υ			
Src TypesD*B,*W,*D*E	st Types 3,*W,*D						
DWord	Bit			Desc	cription		
03	127:64	ImmSource					
		Exists If:	([Imm	nSource][Src1.RegFi	le]=='IMM')		
		Format: EU_INSTRUCTION_SOURCES_REG_IMM					
	127:64	RegSource					
		Exists If:	([Reg	Source][Src1.RegFil	e]!='IMM')		
		Format:	EU_IN	NSTRUCTION_SOUR	RCES_REG_REG		
	63:32	Operand Contr	ols				
		Format:	EU_INS	STRUCTION_OPERA	ND_CONTROLS		
	31:0	Header					



	sh	- Shift Left
	Format:	EU_INSTRUCTION_HEADER



shr - Shift Right							
Source:	EuIsa	a					
Length Bias:	4						
				Des	cription		
Perform component-wise logical right shift with zero insertion of the bits in src0 by the shift count indicated in src1, storing the results in dst. Insert zero bits in the number of MSBs indicated by the shift count.							
src0 and dst	src0 and dst can have different types and can be signed or unsigned.						
Note: For wo	ord and DWo	rd operands	s, the ac	cum	ulators have 33 bit	ts.	
Note: For un	signed src0 t	ypes, shr an	d asr pr	odu	ce the same result		
The shift cour unsigned inte	nt is taken fro eger in the ra	om the low f nge 0 to 31.	five bits	of s	rc1, regardless of t	the src1 type and treated as an	
Format:							
[(pred)] shr[.	cmod] (exec_	size) dst src	0 src1				
					Syntax		
[(pred)] sh	ur[.cmod] (exec_size)	reg r	eg	reg [(pred)] sh	r[.cmod] (exec_size) reg reg :	imm32
Evaluate(Wr = src1.char src0.chan[r	En); for ([n] & 0x1F] >> shift	n = 0; n ; // Alway Cnt; } }	< exec /s use	Pseu _si low	ze; n++) { if 5 bits for shi	(WrEn.chan[n]) { shiftCnt ft count. dst.chan[n] =	
Predication	Conditional	Modifier 9	Saturati	ion	Source Modifier		
Y	Y	N	Y		Υ		
Src TypesIUB,UW,UDU	Ost Types IB,UW,UD						
DWord	Bit				Des	cription	
03	127:64	ImmSourc	e				
		Exists If:	([]	[mm	Source][Src1.RegF	ile]=='IMM')	
		Format:	El	J_IN	ISTRUCTION_SOUF	RCES_REG_IMM	
	127:64	RegSource	9				
	Exists If: ([RegSource][Src1.RegFile]!='IMM')						
		Format:	E	U_IN	ISTRUCTION_SOUF	RCES_REG_REG	
	63:32	Operand C	Controls	S			
		Format:	EU	_INS	STRUCTION_OPERA	AND_CONTROLS	
	31:0	Header					
		Format:		E	U_INSTRUCTION_H	HEADER	



f32to16 - Single Precision Float to Half Precision Float

Source:

Length Bias:

The f32to16 instruction converts the single precision float in src0 to half precision float and storing in the lower word of each channel in dst.

Because this instruction does not have a 16-bit floating-point type, the destination data type must be Word (W).

Format:

[(pred)] f32to16[.cmod] (exec_size) dst src0

EuIsa

4

Restriction

Restriction: The destination must be DWord-aligned and specify a horizontal stride (HorzStride) of 2. The 16-bit result is stored in the lower word of each destination channel and the upper word is not modified.

Restriction: The FP Mode (Single Precision Floating Point Mode in cr0) must be IEEE mode.

Restriction: No accumulator access, implicit or explicit.

Syntax

[(pred)] f32to16[.cmod] (exec_size) reg reg [(pred)] f32to16[.cmod] (exec_size) reg imm32

Pseudocode	
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { ds convert single precision float to half precision float(src0.chan[n]); } }</pre>	st.chan[n] =

Predication	Conditional	Modifier S	aturation	Source Modifier			
γ	Y	Y	,	Υ			
Src Types D	st Types						
DWord	Bit			Desc	cription		
03	127:64	ImmSource	•				
		Exists If:	([Opera	([Operand Controls][Src0.RegFile]=='IMM')			
		Format:	EU_INS	EU_INSTRUCTION_SOURCES_IMM32			
	127:64	RegSource					
		Exists If:	([Oper	([Operand Controls][Src0.RegFile]!='IMM')			
		Format:	EU_INSTRUCTION_SOURCES_REG				
	63:32	Operand Co	perand Controls				
		Format:	EU_INS	EU_INSTRUCTION_OPERAND_CONTROLS			
	31:0	Header					
		Format:	E	U_INSTRUCTION_H	IEADER		



SRC_COPY_BLT

Source: BlitterCS

2

Length Bias:

This BLT instruction performs a color source copy where the only operands involved is a color source and destination of the same bit width.

The source and destination operands may overlap. The command must indicate the horizontal and vertical directions: either forward or backwards to avoid data corruption. The X direction (horizontal) field applies to both the destination and source operands. The source and destination pitches (stride) are signed.

DWord	Bit		D	escription			
0	31:29	Client	Client				
DDCC		Default Value:		02h 2D Proces	ssor		
BR00		Format: Opcode					
	28:22	Instruction Target(Op	ocode)				
		Default Value:			43h		
		Format:			Opcode		
	21:20	32bpp Byte Mask					
		This field is only used f	or 32bpp.				
		Value		Na	ame		
		1xb	Write Alpha Cha	nnel			
		x1b	Write RGB Chanr	nel			
	19:6	Reserved					
		Format:		١	MBZ		
	5:0	DWord Length					
		Default Value:			04h		
1	31	Reserved					
DD1 2		Format:		١	MBZ		
BR13	30	X Direction					
		(1 = written from right	to left (decrement	nting = backwa	ards); 0 = incrementing)		
	29:26	Reserved					
		Format:		١	MBZ		
	25:24	Color Depth					
		Value		Ν	lame		
		00b	8 Bit Color				
		01b	16 Bit Color(56	55)			
		10b	16 Bit Color(15	555)			
		11b	32 Bit Color				
	23:16	Raster Operation					



		SRC	COPY_BLT				
	15:0	Destination Pitch (sign Destination pitch in byte	Destination Pitch (signed) Destination pitch in bytes (Same as before).				
2	31:16	Destination Height (in	scan lines)				
BR14	15:0	Destination Byte Widt	Destination Byte Width (in bytes)				
3	31:0	Destination Address					
		Format:	GraphicsAddress[31:0]				
BR09		Address of the first byte	to be written.				
4	31:16	Reserved					
5511		Format:		MBZ			
BR11	15:0	Source Pitch (double word aligned ar	Source Pitch (double word aligned and signed)				
5	31:0	Source Address					
		Format:	Format: GraphicsAddress[31:0]				
BR12		Address of the first byte	to be read.				



STATE_BASE_ADDRESS

Source: BSpec

Length Bias:

The STATE_BASE_ADDRESS command sets the base pointers for subsequent state, instruction, and media indirect object accesses by the GPE. (See Table 4-3. Base Address Utilization for details)

Programming Notes

The following commands must be reissued following any change to the base addresses

- 3DSTATE_CC_POINTERS
- 3DSTATE_BINDING_TABLE_POINTERS

2

- 3DSTATE_SAMPLER_STATE_POINTERS
- 3DSTATE_VIEWPORT_STATE_POINTERS
- MEDIA_STATE_POINTERS

Execution of this command causes a full pipeline flush, thus its use should be minimized for higher performance

DWord	Bit			Description				
0	31:29	Command	Туре					
		Default Val	ue:	3	h GFX	(PIPE		
	28:27	Command	SubType					
		Default Val	ue:	0h GFXPIPE_COM	IOMN	N		
	26:24	3D Comma	nd Opcode					
		Default Val	ue:	1h GFXPIPE_NONPIP	ELINE	D		
	23:16	3D Comma	nd Sub Opcode					
		Default Val	ue:	01h STATE_BASE_AD	DRES	S		
-	15:8	Reserved						
		Format:			Ν	1BZ		
	7:0	DWord Len	gth					
		Format:	=	n Total Length - 2				
		Value		Name		Description		
		8h	DWORD_COUNT_n	[Default]		Excludes DWord (0,1)		
1	31:12	General Sta	te Base Address			1		
		Format:	Graph	nicsAddress[31:12]				
		Specifies the 4K-byte aligned base address for general state accesses. See Table 4-3 for details						
		on where this base address is used.						
	11.0	Conoral Sta	to Momory Object	Control State				
	11.0	Eormat:			c			
		Format.		DJECT_CONTROL_STAT		using the Consul State Rase Address		
		with the exc	e memory object col	ess data port accesses.	tate u	ising the General State Base Address,		
1	15:8 7:0 31:12 11:8	Reserved Format: DWord Len Format: Value 8h General Sta Format: Specifies the on where the General Sta Format: Specifies the with the exc	gth = DWORD_COUNT_n te Base Address Graph e 4K-byte aligned ba is base address is us te Memory Object MEMORY_O e memory object con- reption of the statele	n Total Length - 2 Name [Default] nicsAddress[31:12] use address for general state address for general state BJECT_CONTROL_STATE ntrol state for indirect state ses data port accesses.	state E tate u	Description Excludes DWord (0,1) accesses. See Table 4-3 for de using the General State Base A		



			STAT	E_BASE_ADD	RE	SS				
	7:4	Stateless Data Port Access Memory Object Control State								
		Format: MEMORY_OBJECT_CONTROL_STATE								
		Specifies the	memory object	control state for stateles	s dat	a port ac	ccesses.			
	3	Stateless Dat	ta Port Access	Force Write Thru						
		Format:						11.1		
		0: If the state	eless data port	caccess memory object	t cor	ntrol inc	licates L3 cach	hable the		
		1. If the stat	eless data port	access memory object	-t cor	atrol inc	licates 13 cach	hable the		
		accesses wil	l be write thru	cacheable.						
	2:1	Reserved								
		Format:				MBZ				
	0	General Stat	e Base Address	Modify Enable						
		Format:		E	inable	9				
		The other fields in this dword are updated only when this bit is set.								
		Value	Name	Dese	cripti	ion				
		0h	Disable Ignore the updated address							
		1h	Enable	Modify the address						
2	31:12	Surface State	e Base Address							
		Format:	Gr	aphicsAddress[31:12]						
		Specifies the	4K-byte aligned	base address for bindin	g tab	ole and su	urface state acc	esses. See		
		Table 4-3 for	details on when	e this base address is use	ea.					
	11:8	Surface State	e Memory Obie	ect Control State						
		Format:	MEMORY	_OBJECT_CONTROL_STA	TE					
		Specifies the	memory object	control state for indirect	state	e using tl	he Surface Sta	te Base		
		Address.								
	7.1	December								
	/.1	Format:				MB7				
		Curface Stat	- Pasa Address	Madify Enable		IVIDZ				
	0	Format	e base Address		nahle	<u>ــــــــــــــــــــــــــــــــــــ</u>				
		The other fiel	ds in this dword	ے ا l are updated only when	this I	- hit is set				
		Value	Name	Desc	cripti	ion				
		0h	Disable	Ignore the updated add	lress	-				
		1h	Enable	Modify the address						
3	31.12	Dynamic Sta	te Base Addres	,						
5	51.12	Format:	Gr	aphicsAddress[31:12]						
			0.	· · · · · · · · · · · · · · · · · · ·						



			STAT	E_BASE_ADDRE	SS						
		Specifies the 3 for details of	Specifies the 4K-byte aligned base address for sampler and viewport state accesses. See Table 4 3 for details on where this base address is used.								
	11:8	Dynamic State Memory Object Control State									
		Format: MEMORY_OBJECT_CONTROL_STATE									
Specifies the memory object control state for indirect state using the Dynamic Stat Address . Push constants defined in 3DSTATE_CONSTANT_(VS GS PS) commands this control state, although they can use the corresponding base address. The memory control state for push constants is defined within the command.											
	7:1	Reserved									
		Format:			MBZ						
	0	Dynamic Sta	te Base Addres	ss Modify Enable	-						
		Format:		Enable	9						
		The other fiel	ds in this dword	are updated only when this l	bit is set.						
		Value	Name	Descript	ion						
		0h	Disable	Ignore the updated address							
		1h	Enable	Modify the address							
4	31:12	Indirect Obj	ect Base Addre	SS							
		Format: GraphicsAddress[31:12]									
		Specifies the See Table 4-3	4K-byte aligned for details on v	l base address for indirect obj where this base address is use	ect load in MEDIA_OBJI d.	ECT command.					
	11:8	Indirect Obj	ect Memory Ob	ject Control State							
		Format:	MEMORY	_OBJECT_CONTROL_STATE							
		Specifies the Address .	memory object	control state for indirect obje	cts using the Indirect C)bject Base					
	7:1	Reserved									
		Format:			MBZ						
	0	Indirect Obj	ect Base Addre	ss Modify Enable							
		Format:		Enable	9						
		The other fiel	ds in this dword	are updated only when this	bit is set.						
	Value Name Description										
		0h	Disable	Ignore the updated address							
		1h	Enable	Modify the address							
5	31:12	Instruction E	Base Address								
		Format:	Gr	aphicsAddress[31:12]							
		Specifies the	4K-byte aligned	l base address for all EU instru	action accesses.						
	11:8	Instruction N	Memory Object	Control State							



			STAT	E_BASE_ADDRE	SS				
		Format:	Format: MEMORY_OBJECT_CONTROL_STATE						
		Specifies the Address .	Specifies the memory object control state for EU instructions using the Instruction Base Address.						
	7:1	Reserved							
		Format:			MBZ				
	0	Instruction B	Base Address M	lodify Enable					
		Format:		Enab	e				
		The other fiel	ds in this dword	d are updated only when this	bit is set.				
		Value	Name	Descrip	tion]			
		0h	Disable	Ignore the updated address					
		1h	Enable	Modify the address					
6	31:12	General Stat	e Access Upper	r Bound		<u> </u>			
		Format:	Gr	aphicsAddress[31:12]					
	11:1	General State accesses. This includes an accesses that are offset from General State Base Address (see Table 4-3). Read accesses from this address and beyond will return UNDEFINED values. Data port writes to this address and beyond will be "dropped on the floor" (all data channels will be disabled so no writes occur). Setting this field to 0 will cause this range check to be ignored. If non-zero, this address must be greater than the General State Base Address.							
		Format:			MBZ				
	0	General Stat	e Access Upper	r Bound Modify Enable					
		Format:		Enab	е				
		The bound in	this dword is u	pdated only when this bit is a	set.				
		Value	Name	Descrip	tion				
		0N 16		Ignore the updated bound					
			Enable						
7	31:12	Dynamic Sta	te Access Uppe	er Bound					
Format: [GraphicsAddress[31:12] Specifies the 4K-byte aligned (exclusive) maximum Graphics Memory add dynamic state accesses. This includes all accesses that are offset from Dyn Base Address (see Table 4-3). Read accesses from this address and beyon UNDEFINED values. Data port writes to this address and beyond will be "c floor" (all data channels will be disabled so no writes occur). Setting this fi					ress for amic State Ind will return Iropped on the eld to 0 will				
	cause this range check to be ignored.								



			STAT	E_BASE_ADE	DRE	SS		
		If non-zero,	this address m	ust be greater than t	the Dy	namic State Base A	ddress.	
	11.1	Percentrad						
		Format:				MBZ		
	0	Dynamic Sta	te Access Uppe	r Bound Modify Enab	ble			
	Ũ	Format:		<u> </u>	Enable	2		
		The bound in	this dword is up	odated only when this l	bit is se	et.		
		Value	Name	De	escripti	ion		
		0h	Disable	Ignore the updated b	ound			
		1h	Enable	Modify the bound				
8	31:12	Indirect Obje	ect Access Uppe	er Bound				
		Format:	Gra	aphicsAddress[31:12]				
		This field sp	ecifies the 4K-k	oyte aligned (exclusiv	/e) ma	ximum Graphics Mer	nory address	
		access by an	indirect objec	t load in a MEDIA_O	BJECT	command. Indirect d	lata accessed	
		at this addre	ess and beyond	d will appear to be 0.	Settin	ig this field to 0 will c	ause this	
		range check	to be ignored.					
		If non-zero,	this address m	ust be greater than t	the Inc	direct Object Base A	ddress.	
		Hardware ig	nores this field	l if indirect data is no	ot pres	ent.		
		Setting this	field to FFFFFN	will cause this range	e check	k to be ignored.		
	11:1	Reserved						
		Format:				MBZ		
	0	Indirect Obje	ect Access Uppe	er Bound Modify Enal	ble			
		Format:			Enable	2		
		The bound in	this dword is up	pdated only when this l	bit is se	et.		
		Value	Name	De	escripti	ion		
		0h	Disable	Ignore the updated b	ound			
		1h	Enable	Modify the bound				
9	31:12	Instruction A	ccess Upper Bo	ound				
		Format:	Gra	aphicsAddress[31:12]				
		This field sp	ecifies the 4K-b	oyte aligned (exclusiv	/e) ma	ximum Graphics Mer	mory address	
		access by an	EU instruction	n. Instruction data ac	cessec	at this address and	beyond will	
		return UNDEFINED values. Setting this field to 0 will cause this range check to be						
		ignored.						
		It non-zero,	If non-zero, this address must be greater than the Instruction Base Address .					
	11:1	Reserved						
		Format:				MBZ		
	0	Instruction A	ccess Upper Bo	ound Modify Enable				
		Format:			Enable	2		



STATE_BASE_ADDRESS							
	The bound in this dword is updated only when this bit is set.						
	Value	Name	Description				
	0h	Disable	Ignore the updated bound				
	1h	Enable	Modify the bound				



	STATE_SIP							
Source:	Source: BSpec							
Length E	Length Bias: 2							
The STA	TE_SIP	command	specifies the starting	g ins	struction location of	f the S	System Routine that is shar	red by all
threads	in exec	ution.						
DWord	Bit				Descriptio	on		
0	31:29	Command	d Type			1		
		Default V	alue:			3h G	FXPIPE	
	28:27	Command	d SubType					
		Default V	alue:		0h GFXPIPE_CC	OMMO	NC	
	26:24	3D Comm	and Opcode		1			
		Default V	alue:		1h GFXPIPE_NONP	IPELIN	NED	
	23:16	3D Comm	and Sub Opcode					
		Default V	alue:		02h	STATE	_SIP	
	15:8	Reserved						
		Format:					MBZ	
	7:0	DWord Le	ength					
		Format:		=n	Total Length - 2			
								1
		Value	Ν	lam	le		Description	
		0h	DWORD_COUNT_r	ח [D	efault]	Exclu	udes DWord (0,1)]
1	31:4	System Instruction Pointer						
Format: InstructionBaseOffset[31:4]Kernel								
Specifies the instruction address of the system routine associated with the current con						ntext as a		
	128-bit granular offset from the Instruction Base Address. SIP is shared by all threads in execution. The address specifies the double quadword aligned instruction location						in	
						a ang		
	3:0	Reserved						
Format: MBZ								



sad2 - Sum of Absolute Difference 2

Source: EuIsa

Length Bias:

The sad2 instruction takes source data channels from src0 and src1 in groups of 2-tuples. For each 2-tuple, it computes the sum-of-absolute-difference (SAD) between src0 and src1 and stores the scalar result in the first channel of the 2-tuple in dst.

The results are also stored in the accumulator register. The destination operand and the accumulator maintain 16 bits per channel precision.

The destination register must be aligned to even word (DWord). The even words in the destination region will contain the correct data. The odd words are also written but with undefined values.

Format:

[(pred)] sad2[.cmod] (exec_size) dst src0 src1

Restriction

Restriction: Source operands cannot be accumulators.

Restriction: The execution size cannot be 1 as the computation requires at least two data channels.

Syntax

[(pred)] sad2[.cmod] (exec_size) reg reg reg [(pred)] sad2[.cmod] (exec_size) reg reg imm32

Pseudocode

 $Evaluate(WrEn); \ for \ (\ n = 0; \ n < exec_size; \ n += 2 \) \ \left\{ \ if \ (\ WrEn.chan[n] \) \ \left\{ \ dst.chan[n] = abs(src0.chan[n] - src1.chan[n]) \ + \ abs(src0.chan[n+1] - src1.chan[n+1]); \ \right\} \ \right\}$

Predication	Conditional Modifier	Saturation	Source Modifier
V	v	V	V

Src Types Dst Types

D,UD	vv,0vv					
DWord	Bit		Description			
03	127:64	ImmSource				
		Exists If:	([ImmSource][Src1.RegFile]=='IMM')			
		Format:	EU_INSTRUCTION_SOURCES_REG_IMM			
	127:64	RegSource				
		Exists If:	([RegSource][Src1.RegFile]!='IMM')			
		Format:	EU_INSTRUCTION_SOURCES_REG_REG			
	63:32	Operand Cor	ntrols			
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS			
	31:0	Header	Header			
		Format:	EU_INSTRUCTION_HEADER			



sada2 - Sum of Absolute Difference Accumulate 2

Source: EuIsa Length Bias: 4

The sada2 instruction takes source data channels from src0 and src1 in groups of 2-tuples. For each 2-tuple, it computes the sum-of-absolute-difference (SAD) between src0 and src1, adds the intermediate result with the accumulator value corresponding to the first channel, and stores the scalar result in the first channel of the 2-tuple in dst.

The destination operand and the accumulator maintain 16 bits per channel precision. Higher precision (guide bits) stored in the accumulator allows up to 64 rounds of sada2 instructions to be issued back to back without overflowing the accumulator.

The destination register must be aligned to even word (DWord). The even words in the destination region will contain the correct data. The odd words are also written but with undefined values.

Format:

[(pred)] sada2[.cmod] (exec_size) dst src0 src1

Restriction

Restriction: Source operands cannot be accumulators.

Restriction: The execution size cannot be 1 as the computation requires at least two data channels.

Syntax [(pred)] sada2[.cmod] (exec_size) reg reg reg [(pred)] sada2[.cmod] (exec_size) reg reg imm32

Pseudocode

 $Evaluate(WrEn); \ for \ (\ n = 0; \ n < exec_size; \ n += 2 \) \ \left\{ \begin{array}{l} uwTmp = abs(src0.chan[n] - src1.chan[n]) + abs(src0.chan[n+1] - src1.chan[n+1]); \ if \ (\ WrEn.chan[n] \) \ \left\{ \begin{array}{l} dst.chan[n] = uwTmp + acc[n]; \ \end{array} \right\}$

Predication	Conditional Modifier	Saturation	Source Modifier
Y	Υ	Y	Υ

Src Types Dst Types

B,UB W,UW

DWord	Bit	Description			
03	127:64	ImmSource			
		Exists If:	([ImmSource][Src1.RegFile]=='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_REG_IMM		
	127:64	RegSource			
		Exists If:	([RegSource][Src1.RegFile]!='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_REG_REG		
	63:32	Operand Cont	Operand Controls		
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS		



sada2 - Sum of Absolute Difference Accumulate 2					
	31:0	Header			
		Format:	EU_INSTRUCTION_HEADER		



SWTESS_BASE_ADDRESS

Source: BSpec

Length Bias:

The SWTESS_BASE_ADDRESS command sets the base pointers for SW Tessellation data read access by the TE unit.

Programming Notes

This base address must also be comprehended in the SURFACE_STATE used by the HS kernel to write the SW tessellation data.

Execution of this command causes a full pipeline flush, thus its use should be minimized for higher performance.

DWord	Bit	Description					
0	31:29	Command	Туре				
		Default Val	ue:		3h GFX	PIPE	
	28:27	Command	SubType				
		Default Val	ue:	0h GFXPIPE_C	OMMO	٨	
	26:24	3D Comma					
		Default Value: 1h GFXPIPE_NONPIPELINED				D	
	23:16	3D Comma	nd Sub Opcode				
		Default Val	ue:	03h SWTESS_BASE	_ADDRES	SS	
	15:8	Reserved					
		Format:			М	BZ	
	7:0	DWord Length					
		Format:	=	=n Total Length - 2			
		Value		Name		Description	
		0h	DWORD_COUNT_r	n [Default]		Excludes DWord (0,1)	
1	31:12	SW Tessella	ation Base Address	s			
		Format:	Grap	hicsAddress[31:12]			
		Specifies the 4K-byte aligned base address for TE unit SW tessellation data read accesses.					
	11:8	SW Tessella	ation Memory Obj	ect Control State			
		Format:	MEMORY_C	DBJECT_CONTROL_ST	ATE		
		Specifies the memory.	e memory object co	ontrol state used by t	he TE un	it to read SW tessellation data from	
	7:0	Reserved					
		Format:			М	BZ	



		W	ait - W	ait Notifie	cation		
Source:	Ει	ıIsa					
_ength Bias	4						
The wait instruction evaluates the value of the notification count register nreg. If nreg is zero, thread execution is suspended and the thread is put in 'wait_for_notification' state. If nreg is not zero (i.e., one or more notifications have been received), nreg is decremented by one and the thread continues executing on the next instruction. If a thread is in the 'wait_for_notification' state, when a notification arrives, the notification count register is incremented by one. As the notification count register becomes nonzero, the thread wakes up to continue execution and at the same time the notification register is decremented by one. If only one notification arrived, the notification register value becomes zero. However, during the above mentioned time period, it is possible that more notifications may arrive, making the notification register nonzero again.							
registers fo	r each notifi	cation.					
Notificatio and n1:ud thread-thre Format:	Notification register n0:ud is for thread to thread communication (via the Message Gateway shared function) and n1:ud for host to thread communication (through MMIO registers). See the Message Gateway chapter for thread-thread communication. Format:						
	size) fileg						
				Restriction			
Restriction	src0 and ds	t must be n0,	n1, or n2.				
Restriction	Execution s	ize must be 1	as the notifi	cation registers are	scalar.		
Restriction	Predication	is not allowed	d.				
Restriction: inserted be	Two back-te tween two v	o-back wait in vait instruction	structions ar	e not allowed. At r	ninimum, a nop instruction must be		
				Syntax			
wait (1)	n#			Syntax			
ware (r)							
				Pseudocode			
N/A							
Predicatio	n Conditio	nal Modifier	Saturation	Source Modifier			
N	Ν		Ν	N			
Src Types	Dst Types						
UD	UD	D					
DWord	Bit		Description				
0	127:64	Sources	Sources				
		Exists If:	([Oper	and Control][Src1.F	RegFile]=='IMM')		
		Format:	EU_INS	STRUCTION_SOUR	CES_IMM32		
	127:64	Sources	ources				



wait - Wait Notification					
		Exists If:	([Operand Control][Src1.RegFile]!='IMM')		
		Format:	EU_INSTRUCTION_SOURCES_REG		
	63:32	Operand Control			
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
	31:0	Header			
		Format:	EU_INSTRUCTION_HEADER		



while - While								
Source:	Source: EuIsa							
Length Bia	Length Bias: 4							
			Des	cription				
The while termination flags spect address s continues of code. If dst and m	The while instruction marks the end of a do-while block. The instruction first evaluates the loop termination condition for each channel based on the current channel enables and the predication flags specified in the instruction. If any channel has not terminated, a branch is taken to a destination address specified in the instruction, and the loop continues for those channels. Otherwise, execution continues to the next instruction. Id point to the first instruction with the do label of the do-while block of code. It should be a negative number for the backward referencing. In GEN binary, JIP is at location dst and must be of type W (signed word integer).							
If SPF is 0	ON, nor	e of the PcIP are up	dated.					
The follow IP pre-inc code. It sh src1 and r	The following table describes the 16-bit jump target offset JIP. JIP is a signed 16-bit number, added to IP pre-increment, and should point to the first instruction with the do label of the do-while block of code. It should be a negative number for the backward referencing. In GEN binary, JIP is at location src1 and must be of type W (signed word integer).							
Format: [(pred)] v	vhile (ex	ec_size) JIP						
				Restriction				
Restriction: The execution size must be the same for the while instruction and any break and cont instructions of the same code block.								
Syntax								
[(pred)]	[(pred)] while (exec_size) imm16							
Pseudocode								
<pre>Evaluate(WrEn); for (n = 0; n < 32; n++) { if (WrEn.chan[n]) { PcIP[n] = IP + JIP; } else { PcIP[n] = IP + 1; } } if (PMask == 1) { // any enabled channel true Jump(IP + JIP); }</pre>								
Predicati	on Cor	nditional Modifier	Saturation	Source Modifier				
Υ	Ν		N	Ν				
DWord	Bit	Description						
03 1	27:112	Reserved						
		Format: MBZ						
1	111:96	JIP						
		Format:			S15			
		Jump Target Offset	. The relative	e offset in 64-bit un	hits if a jump is taken for the instruction.			
	95:91	Reserved						



while - While

	[
	Format:			MBZ			
90	90 Flag Register Number						
	Added a second flag register						
89	Flag Subr	egister Nur	nber				
	This field specifies the sub-register number for a flag register operand. There are two sub-						
	registers in the flag register. Each sub-register contains 16 flag bits.						
	Ine selected flag sub-register is the source for predication if predication is enabled for the						
	for the instruction. The same flag sub-register can be both the predication source and						
	conditional destination, if both predication and conditional modifier are enabled.						
88:64	Source 0						
	Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align1')					
	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1					
88:64	Source 0						
	Exists If:	(Structure	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]=='Align16')				
	Format:	ormat: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16					
63:32	Operand Control						
	Format: EU_I		NSTRUCTION_OPERAND_CONTROLS				
31:0	Header						
	Format:		EU_INSTRUCTION_HEADER				



XY_COLOR_BLT

Source: BlitterCS

Length Bias:

COLOR_BLT is the simplest BLT operation. It performs a color fill to the destination (with a possible ROP). The only operand is the destination operand which is written dependent on the raster operation. The solid pattern color is stored in the pattern background register.

This instruction is optimized to run at the maximum memory write bandwidth.

The typical (and fastest) Raster operation code = F0 which performs a copy of the pattern background register to the destination.

DWord	Bit	Description							
0	31:29	Client							
		Default Value:			02h 2	02h 2D Processor			
BK00		Format:			Орсо	de			
	28:22	Instruction Target(Opcode)							
		Default Valu	e:		50h				
		Format:					Opcode	е	
	21:20	32bpp Byte Mask This field is only used for 32bpp.							
		Valu	e			Na	me		
		1xb	Write Alpha Cha	ite Alpha Channel					
		x1b Write RGB Channel			nel	ا			
19:12 Reserved									
		Format: MBZ							
11 Tiling Enable Des									
								Description	
0b Tiling Disabled (Linear Blit))					
		1bTiling EnabledTile-X or					or Tile-Y.		
	10:8	Reserved Format: MBZ							
7:0 DWord Length						1			
		Default Value: 04h							
1	31	Reserved							
DD1 2		Format: MBZ							
DUID	30 Clipping Enabled								
			Value Name						
		0b	D Disabled						



		X	Y_COLOR_BLT				
		1b Enabled					
	29:26	Reserved					
		Format:	MBZ				
	25:24	Color Depth					
		Value	Name				
		00b	8 Bit Color				
		01b	16 Bit Color(565)				
		10b	16 Bit Color(1555)				
		11b	32 Bit Color				
	23:16	Raster Operation					
	15:0	Destination Pitch in DWords 2's complement For Tiled surfaces (bit_11 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).					
2	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.					
BR22	15:0	Destination X1 Coordinate (Left) 16 bit signed number.					
3	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.					
BR23	15:0	Destination X2 Coordinate (Right) 16 bit signed number.					
4	31:0	Setup Destination Base Address					
DD OO		Format:	GraphicsAddress[31:0]				
БКОЭ		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.					
5 BR16	31:0	Solid Pattern Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]					



XY	FUI	BLT

Source: BlitterCS 2

Length Bias:

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The source and pattern operands are the same bit width as the destination operand.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is less than Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is less than Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

DWord	Bit	Description					
0	31:29	Client					
		Default Value:		02h 2D Processor			
BR00		Format:		Opcode			
	28:22	Instruction Target(Op	code)				
		Default Value:			55h		
		Opcode					
	This field is only used for 32bpp.						
		Value Name					
00b [Default]							
		annel					
		x1b	Write RGB Channel				
Format: MBZ					MBZ		
	15	Src Tiling Enable					


				XY_FUL	L_BLT					
		Value		Nan	ne	Description				
		0b	Tiling Disab	led (Linear Blit	z)					
		1b	Tiling Enabl	ed		Tile-X or Tile-Y.				
	14:12	Pattern Hor Pixel of the s								
	11	Dest Tiling Enable								
		Value		Nan	ne	Description				
		0b	Tiling Disab	led (Linear Blit	t)					
		1b	Tiling Enabl	ed		Tile-X or Tile-Y.				
10:8 Pattern Vertical Seed Starting scan line of the 8x8 pattern corresponding to DST Y=0.						Y=0.				
	7:0	DWord Leng	yth							
		Default Valu	ie:	e: 07h						
1	31	Reserved								
0010		Format:			Μ	IBZ				
BK13	30	Clipping Ena	abled							
		Value			Name					
		0b			Disabled					
		1b Enabled								
	29:26	Reserved								
		Format:	Format: MBZ							
	25:24	Color Depth								
		Va	lue		N	ame				
		00b		8 Bit Color						
		01b		16 Bit Color(5	(565)					
		10b		16 Bit Color(1	1555)					
		11b		32 Bit Color						
	23:16	Raster Oper	ation							
	15:0	Destination 2's complem X, 128B gran	Pitch in DW ent For Tiled ularity for Tile	ords surfaces (bit_1 e-Y and can be	.1 enabled) this pitc e upto 128Kbytes (c	ch is of 512Byte granularity for Tile- or 32KDwords).				
2	31:16	Destination	Y1 Coordina	te (Top)						
		16 bit signed	number.							
BRZZ	15:0	Destination	X1 Coordina	ite (Left)						
2	21.10	16 bit signed	Na Coordina							
3	31:10	16 bit signed	12 Coordina I number	ite (Bottom)						
BR23	15:0	Destination	X2 Coordina	te (Right)						
		16 bit signed	number.	· · · · · · · · · · · · · · · · · · ·						



		XY_FULL_BLT						
4	31:0	Destination Base Address						
		Format: GraphicsAddress[31:0]						
BR09		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.						
5	31:16	Reserved						
		Format: MBZ						
BR11		Should be programmed all 0's for 48bit addressing.						
	15:0	Source Pitch (double word aligned and signed) and in DWords 2's complement. For Tiled Src (bit 15 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).						
6	31:16	Source Y1 Coordinate (Top) 16 bit signed number.						
BR26	15:0	Source X1 Coordinate (Left) 16 bit signed number.						
7	31:0	Source Address						
		Format: GraphicsAddress[31:0]						
BR12		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_15 enabled), this address is limited to 4Kbytes.						
8 BR15	31:0	Pattern Base (28:06 are implemented) (Note no NPO2 change here). The pattern data must be located in linear memory.						



XY_FULL_IMMEDIATE_PATTERN_BLT

Source: BlitterCS Length Bias: 2

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The source and immediate pattern operands are the same bit width as the destination operand. The immediate data sizes are 64 bytes (16 DWs), 128 bytes (32 DWs), or 256 (64 DWs) for 8, 16, and 32 bpp color patterns. DWL indicates the total number of Dwords of immediate data.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is less than Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is less than Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

DWord	Bit	Description							
0	31:29	Client							
		Default Value:		02h 2D Processor					
BR00		Format:		Opcode					
	28:22	Instruction Target(Ope	nstruction Target(Opcode)						
		Default Value:	Default Value: 74h						
		Format: Opcode							
	21:20	32bpp Byte Mask This field is only used for 32bpp							
		Value		Na	ame				
		00b	[Default]						
		1xb Write Alpha Channel							
		x1b	Write RGB Channe	el					
	19:16	Reserved							



XY_FULL_IMMEDIATE_PATTERN_BLT

			_							
		Format:				MBZ				
	15	Src Tiling Enable								
		Value		Nan	ne		Description			
		0b	Tiling Disal	oled (Linear)						
		1b	Tiling Enab	led		Т	īle-X or Tile-Y.			
	14:12	Pattern Horizontal Seed (pixel of the scan line to start on corresponding to DST X=0)								
	11	Dest Tiling Enable								
		Value		Nar	ne		Description			
		0b	Tiling Disab	led (Linear Bli	t)					
		1b	Tiling Enabl	ed			Tile-X or Tile-Y.			
	10:8	Pattern Vertical Seed Starting scan line of the 8x8 pattern corresponding to DST Y=0.								
	7:0	DWord Lengt	h							
		Default Value: 06h Excludes DWORD 0,1								
		06 + DWL = (I	DWL = (Number of Immediate double words)h							
1	31	Reserved	Reserved							
RD12		Format:		MBZ						
BR13	30	Clipping Enabled								
			Value				Name			
		0b			Disabled					
		1b Enabled								
	29:26	Reserved								
		Format: MBZ								
	25:24	Color Depth								
		Valu	ue			Name				
		00b		8 Bit Color						
		01b		16 Bit Color(565)					
		10b		16 Bit Color(1555)					
		11b		32 Bit Color						
	23:16	Raster Opera	tion							
	15:0	Destination P 2's compleme X, 128B granu	Pitch in DW nt For Tiled larity for Tile	ords surfaces (bit_1 e-Y and can be	.1 enabled) this p e upto 128Kbytes	oitch is o (or 32k	of 512Byte granularity for Tile- (Dwords).			
2	31:16	Destination Y 16 bit signed r	'1 Coordina number.	te (Top)						
BR22	15:0	Destination X 16 bit signed r	(1 Coordina number.	te (Left)						



		XY_FULL_IM	MEDIATE_PATT	ERN_BLT					
3	31:16	Destination Y2 Coordina 16 bit signed number.	Destination Y2 Coordinate (Bottom) L6 bit signed number.						
BR23	15:0	Destination X2 Coordina 16 bit signed number.	te (Right)						
4	31:0	Destination Base Addres	s						
		Format:	GraphicsAddress[31:0]						
BK09		Base address of the destination surface: $X=0$, $Y=0$. When Src Tiling is enabled (Bit_1) this address is limited to 4Kbytes.							
5	31:16	Reserved							
5544		Format:		MBZ					
BR11		Should be programmed all 0's for 48bit addressing.							
	15:0	Source Pitch (double word aligned and signed) and in DWords 2's complement. For Tiled Src (bit 11 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords)							
6	31:16	Source Y1 Coordinate (T 16 bit signed number.	op)						
BR26	15:0	Source X1 Coordinate (Lo 16 bit signed number.	Source X1 Coordinate (Left) 16 bit signed number.						
7	31:0	Source Address							
DD1 2		Format:	GraphicsAddress[31:0]						
BK12		Base address of the destin	ation surface: X=0, Y=0. When	Tiling is enabled (Bit 15 enabled), this					
		address is limited to 4Kbyt	tes.						
8n	31:0	Immediate Data 0							



XY_FULL_MONO_PATTERN_BLT

Source: BlitterCS Length Bias: 2

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The pattern operand is monochrome and the source operand is the same bit width as the destination operand.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is less than Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is less than Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

The monochrome pattern transparency mode indicates whether to use the pattern background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the pattern foreground color is used in the ROP operation.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

Setting both Solid Pattern Select =1 and Mono Pattern Transparency = 1 is mutually exclusive. The device implementation results in NO PIXELs DRAWN.

DWord	Bit	Description						
0	31:29	Client						
		Default Value:	or					
BR00		Format: Opcode						
	28:22	Instruction Target(Opcode)						
		Default Value:		57h				
		Format:		Opcode				
	21:20	32bpp Byte Mask This field is only used for 32bpp.						



XY_FULL_MONO_PATTERN_BLT

		Valu	e	Name ID-G-ski					
		006		[De	efault				
		1xb		Writ	Write Alpha Channel				
		x1b			e RGB Chan	nel			
	19:16	Reserved							
	15	Src Tiling En	able						
		Value			Nam	ıe			Description
		0b	Tiling Dis	abled	l (Linear Blit)			
		1b	Tiling Ena	abled					Tile-X or Tile-Y.
	14:12	Pattern Hori (pixel of the s	tern Horizontal Seed el of the scan line to start on corresponding to DST X=0)						
	11	Dest Tiling E	nable						
Value Name				ne			Description		
		0b	Tiling Dis	abled	l (Linear Blit)			
		1b	Tiling Ena	abled					Tile-X or Tile-Y.
	10:8	Pattern Vect Starting scan	ical Seed line of the	e 8x8	8x8 pattern corresponding to DST Y=0.				
	7:0	DWord Lena	th		•	<u> </u>	5		
			Val	lue					Name
		0Ah							
1	31	Solid Pattern	Select						
		Va	lue					Name	
BR13		0		No Solid Pattern					
		1		Solid Pattern					
	30	Clipping Ena	bled						
			Value						Name
		0b			Disabled				
		1b			Enabled				
	29	Reserved							
		Format:						MBZ	
	28:27	Mono Source	e Transpa	rency	/ Mode				
	Value Name								
		0		Use E	Background				
		1		Trans	sparency En	abled			
	26	Reserved							
		Format:						MBZ	
	25:24	Color Depth							



XY_FULL_MONO_PATTERN_BLT

		Value	Name					
		00b	8 Bit Color					
		01b	16 Bit Color(565)					
		10b	16 Bit Color(1555)					
		11b	32 Bit Color					
	23:16	Raster Operation						
	15:0	Destination Pitch in DW 2's complement For Tiled X, 128B granularity for Tile	ords surfaces (bit_11 enabled) this pitch is of 512Byte granularity for Tile- e-Y and can be upto 128Kbytes (or 32KDwords).					
2	31:16	Destination Y1 Coordina 16 bit signed number.	Destination Y1 Coordinate (Top) 16 bit signed number.					
BR22	15:0	Destination X1 Coordina 16 bit signed number.	te (Left)					
3	31:16	Destination Y2 Coordina 16 bit signed number.	te (Bottom)					
BR23	15:0	Destination X2 Coordina 16 bit signed number.	te (Right)					
4	31:0	Destination Base Addres	Destination Base Address					
		Format: GraphicsAddress[31:0]						
BR09		Base address of the destin this address is limited to 4	ation surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), Kbytes.					
5	31:16	Reserved						
5544		Format:	MBZ					
BR11	15:0	Source Pitch (double wo 2's complement. For Tiled 128B granularity for Tile-Y	rd aligned and signed) and in DWords Src (bit 15 enabled) this pitch is of 512Byte granularity for Tile-X, and can be upto 128Kbytes (or 32KDwords).					
6	31:16	Source Y1 Coordinate (T 16 bit signed number.	op)					
BR26	15:0	Source X1 Coordinate (L 16 bit signed number.	eft)					
7	31:0	Source Base Address						
		Format:	GraphicsAddress[31:0]					
BR12		(base address of the source surface: X=0, Y=0). When Src Tiling is enabled (Bit 15 enabled this address is limited to 4Kbytes.						
8 BR16	31:0	Pattern Background Cold 8 bit = [7:0], 16 bit = [15:0	or], 32 bit = [31:0]					
9	31:0	Pattern Foreground Cold 8 bit = [7:0], 16 bit = [15:0	or], 32 bit = [31:0]					
RKT/								



XY_FULL_MONO_PATTERN_BLT							
10	31:0	Pattern Data 0					
		(least significant DW)					
BR20							
11	31:0	Pattern Data 1					
		(most significant DW)					
BR21							



XY_FULL_MONO_PATTERN_MONO_SRC_BLT

Source:

Length Bias:

BlitterCS

2

The full BLT provides the ability to specify all 3 operands: destination, source, and pattern. The pattern and source operands are monochrome.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation.

All non-text monochrome sources are word aligned. At the end of a scan line the monochrome source, the remaining bits until the next word boundary are ignored. The Monochrome source data bit position field [2:0] indicates which bit position within the first byte should be used as the first source pixel which corresponds to the destination X1 coordinate.

The monochrome pattern transparency mode indicates whether to use the pattern background color or de-assert the write enables when the bit in the pattern is 0. When the source bit is 1, then the pattern foreground color is used in the ROP operation. The monochrome source transparency mode works identical to the pattern transparency mode.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

Setting both Solid Pattern Select =1 and Mono Pattern Transparency = 1 is mutually exclusive. The device implementation results in NO PIXELs DRAWN.

5		· · · · ·					
DWord	Bit	Description					
0	31:29	lient					
		Default Value:	02h 2D Processor				
BR00		Format:	Opcode				
	28:22	Instruction Target(Opcode)					

Negative Stride (= Pitch) is NOT ALLOWED.



	Х	Y_FULL		10	PATT	ERN	I_MON	IO _	SRC_BLT	
		Default Value:					58h			
		Format:						Орсо	ode	
	21:20	32bpp Byte Mask This field is only used for 32bpp.								
		Valu	ie				Na	me		
		00b	00b [Default]							
		1xb		Wr	ite Alpha Cha	annel				
		x1b		Wr	ite RGB Chan	inel				
	19:17	Monochrome source data bit position of the first pixel within a byte per scan line.								
	16:15	Reserved Format:								
	14:12	Pattern Horizontal Seed (pixel of the scan line to start on corresponding to DST X=0)								
	11	Tiling Enable	Tiling Enable						1	
		Value Name							Description	
		0b	Tiling Disabled (Linear Blit)							
		1b Tiling Enabled Tile-X or Tile-Y.						Tile-X or Tile-Y.		
	10:8	Pattern Vertical Seed Starting scan line of the 8x8 pattern corresponding to DST Y = 0.								
	7:0	DWord Leng	DWord Length							
			Val	lue		Name			Name	
		0Ah								
1	31	Solid Pattern	Select							
BR13		Va	lue				Name			
21120		0	No Solid Patter			ttern	ern			
					Solid Patteri	n				
	30	Clipping Ena	bled							
		Ob	value			Name				
		00 1b				Disabled				
	20	Mone Service	- T #		v Mede	LIIADI	eu			
	29	Value		renc	y wode		Na	me		
		0	•	Use	Background	Name				
		1		Tran	isparency Ena	abled				
	28	Mono Patter	n Transpa	ren	v Mode					
		Valu	e				Na	me		
		0		Use	Background					
		1		Tran	sparency Ena	abled				



XY_FULL_MONO_PATTERN_MONO_SRC_BLT

	27:26	Reserved						
		Format:	MBZ					
	25:24	Color Depth						
		Value	Name					
		00b	8 Bit Color					
		01b 16 Bit Color(565)						
		10b	16 Bit Color(1555)					
		11b	32 Bit Color					
	23:16	Raster Operation						
	15:0	Destination Pitch in DWords 2's complement For Tiled surfaces (bit_11 enabled) this pitch is of 512Byte granularity for Tile- 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).						
2	31:16	Destination Y1 Coordina 16 bit signed number.	te (Top)					
BR22	15:0	Destination X1 Coordina 16 bit signed number.	estination X1 Coordinate (Left) 6 bit signed number.					
3	31:16	Destination Y2 Coordina 16 bit signed number.	estination Y2 Coordinate (Bottom) 6 bit signed number.					
BR23	15:0	Destination X2 Coordina 16 bit signed number.	Destination X2 Coordinate (Right)					
4	31:0	Destination Base Address	5					
		Format:	GraphicsAddress[31:0]					
BR09		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.						
5	31:0	Mono Source Address						
		Format: GraphicsAddress[31:0]						
BR12		(address corresponds to D	ST X1, Y1) (Note no NPO2 change here).					
6	31:0	Source Background Colo	r					
BR18		8 bit = [7:0], 16 bit = [15:0]], 32 bit = [31:0]					
7	31.0	Source Foreground Color	r					
		8 bit = [7:0], 16 bit = [15:0]], 32 bit = [31:0]					
BR19								
8	31:0	Pattern Background Cold	pr					
BR16		8 bit = $[7:0]$, 16 bit = $[15:0]$], 32 bit = [31:0]					
Q	31.0	Pattern Foreground Colo						
BR17	51.0	8 bit = [7:0], 16 bit = [15:0]], 32 bit = [31:0]					
10	31:0	Pattern Data 0						



	XY_FULL_MONO_PATTERN_MONO_SRC_BLT							
		(least significant DW)						
BR20								
11	31:0	Pattern Data 1						
		(most significant DW)						
BR21								



XY	FULL	MONO	SRC BLT
_	_	-	

Source: BlitterCS

Length Bias:

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The source operand is monochrome and the pattern operand is the same bit width as the destination.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation.

All non-text and non-immediate monochrome sources are word aligned. At the end of a scan line the monochrome source, the remaining bits until the next word boundary are ignored. The Monochrome source data bit position field [2:0] indicates which bit position within the first byte should be used as the first source pixel which corresponds to the Destination X1 coordinate.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

Negative Stride (= Pitch) is NOT ALLOWED

DWord	Bit	Description					
0	31:29	Client		-			
		Default Value:		02h 2D Process	or		
BR00		Format:		Opcode			
	28:22	Instruction Target(Op	Instruction Target(Opcode)				
		Default Value:			56h		
		Format:			Opcode		
	21:20	32bpp Byte Mask This field is only used for 32bpp.					
		Value	Name				
		00b	[Default]				
		1xb	Write Alpha Channel				
		x1b	Write RGB Channe	el			



						SKC_DLI		
	19:17	Monochrome source data bit position of the first pixel within a byte per scan line.						
	16:15	Reserved						
		Format:				MBZ		
	14:12	Pattern Horizontal Seed (pixel of the scan line to start on corresponding to DST X=0)						
	11	Tiling Enable						
		Value		Nan	ne		Description	
		0b	Tiling Disat	oled (Linear Blit	:)			
		1b	Tiling Enab	led			Tile-X or Tile-Y.	
	10:8	Pattern Vert Starting scan	ical Seed line of the 8	8x8 pattern cor	respor	nding to DST Y =	0.	
	7:0	DWord Leng	th	-				
		Value					Name	
		07h						
1	31	Reserved						
0010		Format:				MBZ		
BK13	30	Clipping Enabled						
		Value					Name	
		0b		Disabled				
		1b Enab				abled		
	29	Mono Source Transparency Mode						
		Valu	e			Name		
		0	U	se Background				
		1	Transparency Enabled					
	28:26	Reserved						
		Format: MBZ						
	25:24	Color Depth						
		Val	ue	9 Dit Color		Nam	e	
		000		8 Bit Color				
		10b		16 Bit Color(1				
		11b		32 Bit Color	10 BIL COLOR(1555)			
	22.16	Pactor Opera	tion	52 Dit Coloi				
	25.10 15·0	Destination	Pitch in DW	lords				
	13.0	2's compleme X, 128B granu	ent For Tiled ularity for Til	surfaces (bit_1 e-Y and can be	.1 enal e upto	bled) this pitch is 128Kbytes (or 3	s of 512Byte granularity for Tile- 2KDwords).	
2	31:16	Destination 16 bit signed	Y1 Coordin a number.	ate (Top)				

_ _ _ _

. . . .



		XY_FUL	L_MONO_SRC_BLT				
BR22	15:0	Destination X1 Coordinat 16 bit signed number.	te (Left)				
3	31:16	Destination Y2 Coordinat 16 bit signed number.	e (Bottom)				
BR23	15:0	Destination X2 Coordinat 16 bit signed number.	te (Right)				
4	31:0	Destination Base Address	s				
		Format:	GraphicsAddress[31:0]				
BR09		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.					
5	31:0	Mono Source Address					
		Format:	GraphicsAddress[31:0]				
BR12		(address corresponds to DST X1, Y1) (Note no NPO2 change here).					
6	31:0	Source Background Color 8 bit = [7:0] 16 bit = [15:0]	r 32 hit = [31:0]				
BR18			, 52 510 [5210]				
7	31:0	Source Foreground Color					
		8 bit = [7:0], 16 bit = [15:0]	, 32 bit = [31:0]				
BR19							
8	31:0	Pattern Base Address	1				
		Format:	GraphicsAddress[31:0]				
BKI2		(28:06 are implemented) (Note no NPO2 change here). The pattern data must be located in linear memory.					



XY_FULL_MONO_SRC_IMMEDIATE_PATTERN_BLT

Source: BlitterCS Length Bias: 2

The full BLT is the most comprehensive BLT instruction. It provides the ability to specify all 3 operands: destination, source, and pattern. The source operand is a monochrome and the immediate pattern operand is the same bit width as the destination. The immediate data sizes are 64 bytes (16 DWs), 128 bytes (32 DWs), or 256 (64DWs) for 8, 16, and 32 bpp color patterns.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation.

All non-text monochrome sources are word aligned. At the end of a scan line the monochrome source, the remaining bits until the next word boundary are ignored. The Monochrome source data bit position field [2:0] indicates which bit position within the first byte should be used as the first source pixel which corresponds to the destination X1 coordinate.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

Negative Stride (= Pitch) is NOT ALLOWED.

DWord	Bit	Description					
0	31:29	Client					
		Default Value:		02h 2D Processor			
BR00		Format:		Opcode			
	28:22	Instruction Target(Opc	ode)				
		Default Value:			75h		
		Format:			Opcode		
	21:20 32bpp Byte Mask This field is only used for 32bpp.						
		Value	Name				
		00b	[Default]				
		1xb	Write Alpha Chanr	nel			



	XY_	FULL_N	IONO	_SRC_IM	MEDIATE	_PA	TTERN_BLT		
		x1b		Write RGB Chan	nel				
	19:17	Monochrome source data bit position of the first pixel within a byte per scan line.							
	16:15	Reserved							
		Format:				MBZ			
	14:12	Pattern Hori (pixel of the s	zontal See can line to	d start on correspo	onding to DST X=	=0)			
	11	Tiling Enable							
		Value		Nan	ne		Description		
		0b	Tiling Dis	abled (Linear Blit)	1				
		1b	Tiling Ena	bled			Tile-X or Tile-Y.		
	10:8	Pattern Vert Starting scan	Pattern Vertical Seed Starting scan line of the 8x8 pattern corresponding to DST Y=0.						
	7:0	DWord Leng	th						
		Default Valu	e:	06h E	xcludes DWORD	0,1			
		06 + DWL = (Number o	f Immediate doul	ble words)h				
1	31	Reserved							
0010		Format: MBZ							
DKIS	30	Clipping Enabled							
		Value					Name		
		0b			Disabled				
		1b Enabled							
	29	Mono Source Transparency Mode							
		Valu	e		Name				
		0		Use Background					
		1	Transparency Enabled						
	28:26	Reserved							
		Format:				MBZ			
	25:24	Color Depth							
		Va	lue		Name				
		00b		8 Bit Color					
		01b		16 Bit Color(565)					
		10b		16 Bit Color(1555)					
		11b		32 Bit Color					
	23:16	Raster Opera	ation						
	15:0	Destination 2's compleme 128B granula	Pitch in D ent For Tile rity for Tile	Nords d surfaces (bit_11 -Y and can be up	L enabled) this pi to 128Kbytes (or	tch is of [.] 32KDw	f 512Byte granularity for Tile-X, vords).		



	XY_	FULL_MONO_	SRC_IMMEDIATE_PATTERN_BLT			
2	31:16	Destination Y1 Coordina	te (Top)			
ררסס		16 bit signed number.				
DKZZ	15:0	Destination X1 Coordina	te (Left)			
		16 bit signed number.				
3	31:16	Destination Y2 Coordina	te (Bottom)			
כרחם		16 bit signed number.				
DK23	15:0	Destination X2 Coordinate (Right)				
		16 bit signed number.				
4	31:0	Destination Base Address				
		Format:	GraphicsAddress[31:0]			
BK09		Base address of the destin	ation surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this			
		address is limited to 4Kby	tes.			
5	31:0	Mono Source Address				
		Format:	GraphicsAddress[31:0]			
BR12		(address corresponds to DST X1, Y1) (Note no NPO2 change here).				
6	31:0	Source Background Color				
		8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]				
BR18						
7	31:0	Source Foreground Color				
		8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]			
BR19						
8n	31:0	Immediate Data				



XY_MONO_PAT_BLT

Source: BlitterCS

Length Bias:

MONO_PAT_BLT is used when we have no source and the monochrome pattern is not trivial (is not a solid color only). The monochrome pattern is loaded from the instruction stream.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

The monochrome pattern transparency mode indicates whether to use the pattern background color or de-assert the write enables when the bit in the pattern is 0. When the pattern bit is 1, then the pattern foreground color is used in the ROP operation.

DWord	Bit	Description						
0	31:29	Client	Client					
		Default Value	e:		02h 2D Processo	or		
BR00		Format:			Opcode			
	28:22	Instruction T	arget(Op	code)				
		Default Value	e:			52h		
		Format:				Орсо	ode	
	21:20	32bpp Byte	Mask					
		This field is o	nly used fo	or 32bpp.				
		Valu	е		Na	me		
		00b		[Default]				
		1xb		Write Alpha Chan	nel			
		x1b		Write RGB Channe	el			
	19:15	Reserved						
		Format:			М	1BZ		
	14:12	Pattern Hori	zontal See	ed				
		Pixel of the so	an line to	start on correspon	ding to DST X=0.	•		
	11	Tiling Enable	liling Enable					
		Value	Name				Description	
		0b	Tiling Dis	abled (Linear Blit)				
		1b	Tiling Ena	bled			Tile-X or Tile-Y.	



		XY	_MONO_	PA	T_BL1	•
	10:8	Pattern Vertical Seed Scan line of the 8x8 pat	tern to start on o	corresp	oonding to	DST Y=0.
	7:0	DWord Length				
		Val	ue			Name
		07h				
1	31	Reserved				
0010		Format:				MBZ
BK13	30	Clipping Enabled				
		Value				Name
		0b		Disab	led	
		1b		Enabl	ed	
	29	Reserved				
		Format:				MBZ
	28	Mono Pattern Transpa	arency Mode			
		Value			1	lame
		0	Use Background			
		1	Transparency En	abled		
	27:26	Reserved				
		Format:				MBZ
	25:24	Color Depth				
		Value				Name
		00b	8 Bit Color			
		01b	16 Bit Color(5	565)		
		10b	16 Bit Color(1	L555)		
		11b	32 Bit Color			
	23:16	Raster Operation				
	15:0	Destination Pitch in D	Words			
		2's complement For Tile	ed surfaces (bit_1	.1 enal	bled) this p	bitch is of 512Byte granularity for Tile-
2	21.10	X, 128B granularity for		e upto	128Kbytes	(or 32KDwords).
Z	31:16	16 bit signed number	nate (Top)			
BR22	15·0	Destination X1 Coordi	inate (Left)			
	20.0	16 bit signed number.				
3	31:16	Destination Y2 Coordinate (Bottom)				
		16 bit signed number.				
BR23	15:0	Destination X2 Coordi	inate (Right)			
	21.0	16 bit signed number.				
4	31:0	Destination Base Add	CraphicsAdd	roccI21		
		Format.	GraphicsAdd	ress[3]	1.0]	



	XY_MONO_PAT_BLT								
BR09		Base address of the destination surface: $X=0$, $Y=0$. When Tiling is enabled (Bit 11 enabled), this address is limited to 4Kbytes.							
5	31:0	Pattern Background Color 8 bit – [7:0] 16 bit – [15:0] 32 bit – [31:0]							
BR16		[5,0]							
6	31:0	Pattern Foreground Color							
BR17		8 bit = $[7:0]$, 16 bit = $[15:0]$, 32 bit = $[31:0]$							
7	31:0	Pattern Data 0							
BR20									
8	31:0	Pattern Data 1							
BR21									



XY_MONO_PAT_FIXED_BLT

Source: BlitterCS

2

Length Bias:

MONO_PAT_FIXED_BLT is used when we have no source and the monochrome pattern is not trivial (is not a solid color only). The monochrome pattern is one of 10 fixed patterns described below. The pattern seeds can still be used with the fixed patterns, creating even more fixed patterns. This eliminates 2 doublewords compared to the XY_MONO_PAT_BLT command packet.

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

The monochrome pattern transparency mode indicates whether to use the pattern background color or de-assert the write enables when the bit in the pattern is 0. When the pattern bit is 1, then the pattern foreground color is used in the ROP operation.

DWord	Bit	Description					
0	31:29	Client					
		Default Value:		02h 2D Process	or		
BR00		Format:		Opcode			
	28:22	Instruction Target(Ope	code)				
		Default Value:			59h		
		Format:			Opcode		
	21:20	32bpp Byte Mask This field is only used for 32bpp.					
		Value		Na	ame		
		00b	[Default]				
		1xb	Write Alpha Char	inel			
		x1b	Write RGB Chann	el			
	19	Reserved					
		Format:		N	/IBZ		
	18:15	Fixed Pattern					
		Value Name					
		0000b	HS_HORIZON	TAL			
		0001b	HS_VERTICAL	HS_VERTICAL			



		3	KY_M	ONO_PA	T_FI		BLT	
		0010b		HS_FDIAGO	HS_FDIAGONAL			
		0011b 0100b		HS_BDIAGO	HS_BDIAGONAL			
				HS_CROSS				
		0101b		HS_DIAGCR	SSS			
		0110b		Reserved				
		0111b		Reserved				
		1000b		Screen Door				
		1001b		SD Wide				
		1010b		Walking Bit	(one)			
		1011b		Walking Zer	0			
		1100b		Reserved				
		1101b		Reserved				
		1110b		Reserved				
		1111b		Reserved				
	14:12	Pattern Hori Pixel of the se	tern Horizontal Seed el of the scan line to start on corresponding to DST X=0.					
	11	Tiling Enable						
		Value		Nan	ne	Descriptio		Description
		0b	Tiling Dis	abled (Linear Blit	oled (Linear Blit)			
		1b	Tiling Ena	bled				Tile-X or Tile-Y.
	10:8	Pattern Vert Scan line of t	ical Seed he 8x8 pat	tern to start on c	orresp	onding to E	DST Y=	0.
	7:0	DWord Leng	ıth					
			Val	ue				Name
		05h						
1	31	Reserved				L.		
0012		Format:				Ν	MBZ	
DUID	30	Clipping Ena	bled					
			Value					Name
		0b	Disabled					
		1b	Enabled					
	29	Reserved						
		Format:		MBZ				
	28	Mono Patter	rn Transpa	rency Mode				
		Valu	е			Na	ame	
		0		Use Background				
		1		Transparency En	abled			
	27:26	Reserved						



XY_MONO_PAT_FIXED_BLT

	1								
		Format:	1	MBZ					
	25:24	Color Depth							
		Value	1	Name					
		00b	8 Bit Color						
		01b	16 Bit Color(565)						
		10b 16 Bit Color(1555)							
		11b	32 Bit Color						
	23:16	Raster Operation							
	15:0	Destination Pitch in DW	ords						
		2's complement For Tiled surfaces (bit_11 enabled) this pitch is of 512Byte granularity for Tile X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).							
2	31:16	Destination Y1 Coordina	te (Top)						
220		16 bit signed number.							
BKZZ	15:0	Destination X1 Coordina 16 bit signed number.	te (Left)						
3	31:16	Destination Y2 Coordina	te (Bottom)						
		16 bit signed number.							
BR23	15:0	Destination X2 Coordina	te (Right)						
4	31:0	Destination Base Addres	S						
		Format:	GraphicsAddress[31:0]						
BR09		Base address of the destin	Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit 11 enabled),						
		this address is limited to 4Kbytes.							
5	31:0	Pattern Background Color							
		8 bit = [7:0], 16 bit = [15:0	bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]						
BR16									
6	31:0	Pattern Foreground Cold)r						
BR17		0 bit = [7.0], 10 bit = [15.0]	y, 52 dit = [51.0]						



XY_MONO_SRC_COPY_BLT

Source: BlitterCS

Length Bias:

This BLT instruction performs a monochrome source copy where the only operands involved is a monochrome source and destination. The source and destination operands cannot overlap therefore the X and Y directions are always forward.

All non-text monochrome sources are word aligned. At the end of a scan line of monochrome source, all bits until the next word boundary are ignored. The monochrome source data bit position field [2:0] indicates the bit position within the first byte of the scan line that should be used as the first source pixel which corresponds to the destination X1 coordinate.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation. The ROP value chosen must involve source and no pattern data in the ROP operation. Negative Stride (= Pitch) is NOT ALLOWED.

DWord	Bit	Description							
0	31:29	Client							
		Default Value:			02h 2D Processor				
BR00		Format:			Opcode				
	28:22	Instruction 1	arget(Op	code)					
		Default Valu	e:			54h			
		Format:				Opcode			
	21:20	32bpp Byte	Mask						
		This field is o	nly used fo	or 32bpp.					
		Valu	ie		Na	Name			
		00b		[Default]					
		1xb		Write Alpha Channel					
		x1b		Write RGB Channe	اد				
·	19:17	Monochrome source data bit position of the first pixel within a byte per scan							
	16:12	Reserved							
		Format:				MBZ			
	11	Tiling Enable							
		Value		Name			Description		
		0b	Tiling Dis	abled (Linear Blit)					
		1b	Tiling Enabled Tile-X or Tile-Y.				or Tile-Y.		
	10:8	Reserved							
		Format:			N	/IBZ			
	7:0	DWord Leng	jth						



		XY_M	ONO_SRC	C_C	OPY_BLT					
		Va	lue		Name					
		06h								
1	31	Reserved								
		Format:			MBZ					
BR13	30	Clipping Enabled								
		Value			Name					
		0b		Disab	led					
		1b		Enabl	ed					
	29	Mono Source Transpa	arency Mode							
		Value			Name					
		0	Use Background							
		1	Transparency Enabled							
	28:26	Reserved								
		Format: MBZ								
	25:24	Color Depth								
		Value		Name						
		00b	8 Bit Color							
		01b	16 Bit Color(50	65)						
		10b	16 Bit Color(1555)							
		11b	32 Bit Color							
	23:16	Raster Operation								
	15:0	Destination Pitch in DWords 2's complement For Tiled surfaces (bit_11 enabled) this pitch is of 512Byte granularity for Tile- X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).								
2	31:16	Destination Y1 Coord 16 bit signed number.	inate (Top)							
BR22	15:0	Destination X1 Coord 16 bit signed number.	inate (Left)							
3	31:16	Destination Y2 Coord 16 bit signed number.	inate (Bottom)							
BR23	15:0	Destination X2 Coord 16 bit signed number.	inate (Right)							
4	31:0	Destination Base Add	ress							
		Format:	GraphicsAddre	ess[31	:0]					
BR09		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.								
5	31:0	Source Address								
		Format:	GraphicsAddre	ess[31	.:0]					



	XY_MONO_SRC_COPY_BLT									
BR12		(address corresponding to DST X1,Y1) (Note no NPO2 change here).								
6	31:0	Source Background Color								
		8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]								
BR18										
7	31:0	Source Foreground Color								
		8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]								
BR19										



XY_MONO_SRC_COPY_IMMEDIATE_BLT

Source: BlitterCS

2

Length Bias:

This instruction allows the Driver to send monochrome data through the instruction stream, eliminating the read latency of the source during command execution.

The IMMEDIATE_BLT data MUST transfer an even number of doublewords and the exact number of quadwords. DWL indicates the total number of Dwords of immediate data.

All non-text monochrome sources are word aligned. At the end of a scan line of monochrome source, all bits until the next word boundary are ignored. The Monochrome source data bit position field [2:0] indicates the bit position within the first byte of the scan line that should be used as the first source pixel which corresponds to the destination X1 coordinate.

The monochrome source transparency mode indicates whether to use the source background color or de-assert the write enables when the bit in the source is 0. When the source bit is 1, then the source foreground color is used in the ROP operation. The ROP value chosen must involve source and no pattern data in the ROP operation.

The monochrome source data supplied corresponds to the Destination X1 and Y1 coordinates.

DWord	Bit	Description					
0	31:29	Client		-			
		Default Value:		02h 2D Process	or		
BR00		Format:		Opcode			
	28:22	Instruction Target(Opd	code)				
		Default Value:			71h		
		Format:			Opcode		
	21:20	32bpp Byte Mask This field is only used for 32bpp.					
		Value		Na	ame		
		00b	[Default]				
		1xb	Write Alpha Chani	nel			
x1b Write RGB Channel							
	within a byte per scan line.						
16:12 Reserved							

Negative Stride (= Pitch) is NOT ALLOWED.



XY_MONO_SRC_COPY_IMMEDIATE_BLT Format: MBZ 11 **Dest Tiling Enable** Value Name Description Tiling Disabled (Linear Blit) 0b 1b **Tiling Enabled** Tile-X or Tile-Y 10:8 Reserved Format: MBZ **DWord Length** 7:0 05h Excludes DWORD 0,1 Default Value: 05 + DWL = (Number of Immediate double words)h 1 31 Reserved Format: MBZ BR13 30 **Clipping Enabled** Value Name 0b Disabled 1b Enabled 29 **Mono Source Transparency Mode** Value Name 0b **Transparency Enabled** 1b Use Background 28:26 Reserved Format: MBZ 25:24 **Color Depth** Name Value 00b 8 Bit Color 01b 16 Bit Color(565) 10b 16 Bit Color(1555) 11b 32 Bit Color 23:16 **Raster Operation** 15:0 **Destination Pitch in DWords** 2's complement For Tiled surfaces (bit_11 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords). 2 31:16 Destination Y1 Coordinate (Top) 16 bit signed number. BR22 15:0 **Destination X1 Coordinate (Left)** 16 bit signed number. 3 31:16 **Destination Y2 Coordinate (Bottom)** 16 bit signed number.



	XY_MONO_SRC_COPY_IMMEDIATE_BLT									
BR23	15:0	Destination X2 Coordina	te (Right)							
		16 bit signed number.								
4	31:0	Destination Base Addres	s							
		Format:	GraphicsAddress[31:0]							
BR09		Base address of the destin address is limited to 4Kby	ation surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this tes.							
5	31:0	Source Background Colo	r							
		8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]							
BR18										
6	31:0	Source Foreground Colo	r							
		8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]							
BR19										
7n	31:0	Immediate Data								



XY_PAT_BLT_IMMEDIATE

Source: BlitterCS

Length Bias:

PAT_BLT_IMMEDIATE is used when there is no source and the color pattern is not trivial (is not a solid color only) and the pattern is pulled through the command stream. The immediate data sizes are 64 bytes (16 DWs), 128 bytes (32 DWs), or 256 (64DWs) for 8, 16, and 32 bpp color patterns.

DWL indicates the total number of Dwords of immediate data. All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

DWord	Bit	Description						
0	31:29	Client						
		Default Value:			02h 2D Processor			
BR00		Format:			Opcode			
	28:22	Instruction T	arget(Opc	ode)				
		Default Value	e:			72h		
		Format:				Орсо	ode	
	21:20	32bpp Byte I This field is or	Mask nly used fo	r 32bpp.				
		Valu	е		Na	me		
		00b		[Default]				
		1xb		Write Alpha Chanr	nel			
		x1b		Write RGB Channe)			
	19:15	Reserved						
		Format:			N	1BZ		
	14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.						
	11	Tiling Enable						
		Value		Name			Description	
		0b	Tiling Disa	abled (Linear Blit)				
		1b	Tiling Ena	bled			Tile-X or Tile-Y.	
	10:8	Pattern Vertical Seed Scan line of the 8x8 pattern to start on corresponding to DST Y=0.						



		ΧΥ_ΡΑ	I_BLI_		AIE			
	7:0	DWord Length						
		Default Value:	03h E	Excludes DWORD	0 0,1			
		03 + DWL = (Number of Immediate double)h						
1	31	Reserved						
		Format:			MBZ			
BR13	30	Clipping Enabled						
		Value			Name			
		0b		Disabled				
		1b		Enabled				
	29:26	Reserved						
		Format:			MBZ			
	25:24	Color Depth						
		Value			Name			
		00b	8 Bit Color					
		01b	16 Bit Color(565)					
		10b	16 Bit Color(1555)					
		11b	32 Bit Color					
	23:16	Raster Operation						
	15:0	Destination Pitch in DWo 2's complement (Negative enabled) this pitch is of 512 upto 128Kbytes (or 32KDw	o rds Pitch Not allo 2Byte granula ords).	wed for Pixel nor rity for Tile-X, 12	r Text) For Tiled surfaces (bit_11 8B granularity for Tile-Y and can be			
2	31:16	Destination Y1 Coordinat 16 bit signed number.	e (Top)					
BR22	15:0	Destination X1 Coordinat 16 bit signed number.	te (Left)					
3	31:16	Destination Y2 Coordinat 16 bit signed number.	e (Bottom)					
BR23	15:0	Destination X2 Coordinat 16 bit signed number.	te (Right)					
4	31:0	Destination Base Address	6					
PDOO		Format:	GraphicsAddr	ress[31:0]				
вкоя		Base address of the destina address is limited to 4Kbyte	ation surface: 2 es.	X=0, Y=0. When	Tiling is enabled (Bit_11 enabled), this			
5n	31:0	Immediate Data						



XY	PAT	BLT
		-

BlitterCS Source: 2

Length Bias:

PAT_BLT is used when there is no source and the color pattern is not trivial (is not a solid color only).

If clipping is enabled, all scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

DWord	Bit	Description							
0	31:29	Client							
		Default Value:			02h 2D Processor				
BROO		Format:			Opcode				
	28:22	Instruction T	arget(Opc	ode)					
		Default Value	e:			51h			
		Format:				Орсо	ode		
	21:20	32bpp Byte I This field is or	Mask alv used fo	r 32bpp.					
		Valu	le		Ν	lame			
		00b		[Default]					
		1xb		Write Alpha Chanr	nel				
		x1b	Write RGB Channe)					
	19:15	Reserved							
		Format:				MBZ			
	14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.							
	11	Tiling Enable							
		Value		Name			Description		
		0b	Tiling Disa	abled (Linear Blit)					
		1b	Tiling EnabledTile-X or Tile-Y.						
	10:8	Pattern Verti Scan line of th	attern Vertical Seed can line of the 8x8 pattern to start on corresponding to DST Y=0.						
	7:0	DWord Leng	th						
		Default Value	e:				04h		



			XY_PA ⁻	T_BLT			
1	31	Reserved					
		Format:			MBZ		
BR13	30	Clipping Enabled					
		Value Name					
		0b		Disabled			
		1b		Enabled			
	29:26	Reserved					
		Format:			MBZ		
	25:24	Color Depth					
		Value			Name		
		00b	8 Bit Color				
		01b	16 Bit Color(5	565)			
		10b	16 Bit Color(1	1555)			
		11b	32 Bit Color				
	23:16	Raster Operation					
	15:0	Destination Pitch in DWords 2's complement (Negative Pitch Not allowed for Pixel nor Text) For Tiled surfaces (bit_11 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).					
2	31:16	Destination Y1 Coordina 16 bit signed number.	te (Top)				
BR22	15:0	Destination X1 Coordina 16 bit signed number.	te (Left)				
3	31:16	Destination Y2 Coordina 16 bit signed number.	te (Bottom)				
BR23	15:0	Destination X2 Coordina 16 bit signed number.	te (Right)				
4	31:0	Destination Base Addres	S				
BB0 0		Format:	GraphicsAdd	ress[31:0]			
BK0à		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.					
5	31:0	Pattern Base Address					
		Format:	GraphicsAdd	ress[31:0]			
RK12		(28:06 are implemented) (I linear memory.	Note no NPO2	change here) . T	he pattern data must be located in		



XY_PAT_CHROMA_BLT

Source: BlitterCS

Length Bias:

PAT_BLT is used when there is no source and the color pattern is not trivial (is not a solid color only).

All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

DWord	Bit	Description					
0	31:29	Client					
		Default Value:			02h 2D Processor		
BK00		Format:			Opcode		
-	28:22	Instruction Target(Opcode)					
		Default Value:			76h		
		Format:			Opcode		
-	21:20	32bpp Byte Mask This field is only used for 32bpp.					
		Value		Name			
		00b		[Default]			
		1xb		Write Alpha Channel			
		x1b		Write RGB Channel			
	19:17	Transparency Range Mode (chroma-key) - Dst Chroma-key modes ONLY (SRC ILLEGAL)					
	16:15	Reserved					
		Format:			MBZ		
	14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.					
	11	Tiling Enable					
		Value		Name			Description
		0b	Tiling Disabled (Linear Blit)				
		1b	Tiling Ena	bled			Tile-X or Tile-Y.
	10:8	Pattern Vertical Seed Scan line of the 8x8 pattern to start on corresponding to DST Y=0.					
	7:0	DWord Length					
		Default Value:			06h		


		XY_P	AT_CH	ROMA_BI	LT			
1	31	Reserved						
		Format:			MBZ			
BR13	30	Clipping Enabled						
		Value			Name			
		0b		Disabled				
		1b		Enabled				
	29:26	Reserved						
		Format: MBZ						
	25:24	Color Depth						
		Value	Value Name					
		00b	8 Bit Color					
		01b	16 Bit Color(565)					
		10b	16 Bit Color(1555)					
		11b	32 Bit Color	Color				
	23:16	Raster Operation						
	15:0	Destination Pitch in DWc 2's complement (Negative enabled) this pitch is of 51 upto 128Kbytes (or 32KDw	estination Pitch in DWords s complement (Negative Pitch Not allowed for Pixel nor Text) For Tiled surfaces (bit_11 nabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be pto 128Kbytes (or 32KDwords).					
2	31:16	Destination Y1 Coordina 16 bit signed number.	te (Top)					
BR22	15:0	Destination X1 Coordina 16 bit signed number.	te (Left)					
3	31:16	Destination Y2 Coordinat	te (Bottom)					
		16 bit signed number.	. ,					
BR23	15:0	Destination X2 Coordina 16 bit signed number.	te (Right)					
4	31:0	Destination Base Address	S					
BBO O		Format:	GraphicsAddr	ess[31:0]				
вкоэ		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit 11 enabled), this address is limited to 4Kbytes.						
5	31:0	Pattern Base Address	1					
		Format:	GraphicsAddr	ess[31:0]				
BKI2		(26:06 are used, other bits are ignored) (Note no NPO2 change here). The pattern data must be located in linear memory.						
6	31:0	Transparency Color Low						
0010		(Chroma-key Low = Pixel C	Greater or Equ	al)				
RK18								



XY_PAT_CHROMA_BLT								
7	31:0	Transparency Color High						
		(Chroma-key High = Pixel Less or Equal)						
BR19								



XY_PAT_CHROMA_BLT_IMMEDIATE

Source: BlitterCS 2

Length Bias:

PAT_BLT_IMMEDIATE is used when there is no source and the color pattern is not trivial (is not a solid color only) and the pattern is pulled through the command stream. The immediate data sizes are 64 bytes (16 DWs), 128 bytes (32 DWs), or 256 (64DWs) for 8, 16, and 32 bpp color patterns.

DWL indicates the total number of Dwords of immediate data. All scan lines and pixels that fall within the ClipRect Y and X coordinates are written. Only pixels within the ClipRectX coordinates and the Destination X coordinates are written using the raster operation.

The Pattern Seeds correspond to Destination X = 0 (horizontal) and Y = 0 (vertical). The alignment is relative to the destination coordinates. The pixel of the pattern used / scan line is the (destination X coordinate + horizontal seed) modulo 8. The scan line of the pattern used is the (destination Y coordinate + vertical seed) modulo 8.

DWord	Bit	Description							
0	31:29	Client	Client						
		Default Value	e:		02h 2D Processo	or			
BR00		Format:			Opcode				
	28:22	Instruction T	arget(Opc	ode)					
		Default Value	5:			77h			
		Format:				Орсо	ode		
	21:20	32bpp Byte I This field is or	Mask nly used fo	r 32bpp.					
		Valu	e		Na	me			
		00b		[Default]					
		1xb		Write Alpha Chan	nel				
		x1b		Write RGB Channe	el				
	19:17	Transparency Range Mode (chroma-key) - Dst Chroma-key modes ONLY (SRC ILLEGAL)							
	16:15	Reserved							
		Format:			M	MBZ			
	14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.							
	11 Tiling Enable								
		Value	Name Descr				Description		
		0b	Tiling Disa	abled (Linear Blit)					
		1b	Tiling Ena	bled			Tile-X or Tile-Y.		



		XY_PAT_CH	ROMA_	BLT_IMN	IEDIATE			
	10:8	Pattern Vertical Seed Scan line of the 8x8 patter	n to start on c	orresponding to	DST Y=0.			
	7:0	DWord Length						
		Default Value:	05h E	Excludes DWORD	0,1			
		05 + DWL = (Number of Ir	nmediate dou	ble)h				
1	31	Reserved						
DD1 2	MBZ							
BK13	30	Clipping Enabled						
		Value			Name			
		0b		Disabled				
		1b		Enabled				
	29:26	Reserved						
		Format:			MBZ			
	25:24	Color Depth						
		Value			Name			
		00b	8 Bit Color	Color				
		01b	16 Bit Color(565)					
		10b	16 Bit Color(1555)					
		11b	32 Bit Color					
	23:16	Raster Operation						
	15:0	Destination Pitch in DW 2's complement For Tiled s X, 128B granularity for Tile	ords surfaces (bit_1 -Y and can be	1 enabled) this p upto 128Kbytes	itch is of 512Byte granularity for Tile- (or 32KDwords).			
2	31:16	Destination Y1 Coordina 16 bit signed number.	te (Top)					
BR22	15:0	Destination X1 Coordina 16 bit signed number.	te (Left)					
3	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.						
BR23	15:0	Destination X2 Coordina 16 bit signed number.	te (Right)					
4	31:0	Destination Base Address	S					
BBO O		Format:	GraphicsAddr	ess[31:0]				
BK09		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes.						
5 BR18	31:0	Transparency Color Low (Chroma-key Low = Pixel C	Greater or Equ	al)				



XY_PAT_CHROMA_BLT_IMMEDIATE								
6	31:0	Transparency Color High (Chroma-key High = Pixel Less or Equal)						
BR19								
7n	31:0	Immediate Data						



XY	PIXE	L BLT	
_	-		

Source: BlitterCS

Length Bias:

The Destination X coordinate and Destination Y coordinate is compared with the ClipRect registers. If it is within all 4 comparisons, then the pixel supplied in the XY_SETUP_BLT instruction is written with the raster operation to (Destination Y Address + (Destination Y coordinate * Destination pitch) + (Destination X coordinate * bytes per pixel)).

ROP field must specify pattern or fill with 0's or 1's. There is no source operand.

Negative Stride (= Pitch) specified in the Setup command is Not Allowed

DWord	Bit	Description						
0	31:29	Client						
		Default Val	02h 2D Processor					
BR00		Format:		Opcode				
	28:22	Instruction	Target(Opcode)					
		Default Val	ue:		24h			
		Format:			Орс	ode		
	21:12	Reserved						
		Format:			MBZ			
	11	Tiling Enable						
		Value	Name			Description		
		0b	Tiling Disabled (Linear Bl	it)				
		1b	Tiling Enabled		Tile-X or Tile-Y.			
	10:8	Reserved						
		Format:			MBZ			
	7:0	DWord Len	ıgth					
		Default Value: 00h						
1	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.						
BR22	15:0	Destination 16 bit signe	X1 Coordinate (Left) d number.					



XY_SCANLINES_BLT

Source:	Blit	terCS						
Length Bias:	2							
All scan lines the ClipRect	and pixels (coordina	s that fall wit tes and the l	hin the ClipRect Y and X Destination X coordinate	coordinates a es are written u	re wr sing	itten. Only pixels within the raster operation.		
The Pattern S relative to th coordinate + coordinate +	Seeds corre e destinati horizonta vertical se	espond to D on coordina l seed) mod eed) modulo	estination X = 0 (horizor tes. The pixel of the pati ulo 8. The scan line of th 8.	ntal) and Y = 0 ern used / sca e pattern used	(vert n line l is th	ical). The alignment is e is the (destination X le (destination Y		
Solid pattern	should us	e the XY_SE	TUP_MONO_PATTERN_S	L_BLT instructi	on.			
ROP field mu	ist specify	pattern or fi	ll with 0's or 1's. There is	no source ope	erand	I.		
DWord	Bit		[Description				
0	31:29	Client						
		Default Value: 02h 2D Process			Sor			
BR00		Format:	Format: Opcode					
	28:22	Instruction						
		Default Val	ue:		25h			
		Format:			Орсо	ode		
	21:15	Reserved						
		Format:		Μ	IBZ			
	14:12	Pattern Ho Pixel of the	rizontal Seed scan line to start on corres	ponding to DST	X=0.			
	11	Tiling Enab	le					
		Value	Name	9		Description		
		0b	Tiling Disabled (Linear Bli	t)				
		1b	Tiling Enabled			Tile-X or Tile-Y.		
	10:8	Pattern Vertical Seed Scan line of the 8x8 pattern to start on corresponding to DST Y=0.						
	7:0	DWord Len	igth					
		Default Value: 01h						
1	31:16	Destination 16 bit signe	Y1 Coordinate (Top) d number.					
BR22	15:0	Destination 16 bit signe	x1 Coordinate (Left) d number.					
2	31:16	Destination 16 bit signe	Y2 Coordinate (Bottom) d number.					



XY_SCANLINES_BLT							
BR23	15:0	Destination X2 Coordinate (Right) 16 bit signed number.					



XY_SETUP_BLT

Source: BlitterCS

2

Length Bias:

This setup instruction supplies common setup information including clipping coordinates used by the XY commands: XY_PIXEL_BLT, XY_SCANLINE_BLT, XY_TEXT_BLT, and XY_TEXT_BLT_IMMEDIATE.

These are the only instructions that require that state be saved between instructions other than the Clipping parameters. There are 5 dedicated registers to contain the state for the 3 setup BLT instructions (XY_SETUP_BLT, XY_SETUP_MONO_PATTERN_SL_BLT, and XY_SETUP_CLIP_BLT. All other BLTs use a temporary version of these. The 5 double word registers are: DW1 (Setup Control), DW6 (Setup Foreground color), DW5 (Setup Background color), DW7 (Setup Pattern address), and DW4 (Setup Destination Base Address).

DWord	Bit			I	Description			
0	31:29	Client						
		Default Value:			02h 2D Processor			
BR00		Format:			Opcode			
	28:22	Instruction	Target(Opcode)				
		Default Val	lue:			01h		
		Format:				Орс	code	
	21:20	32 bpp Byt	e Mask					
		Valu	le		Na	ame		
		1xb		Write Alpha C	hannel			
		x1b		Write RGB Cha	annel			
19:12 Reserved								
		Format:			٩	MBZ		
	11	Tiling Enab	le					
		Value		Nam	e	Description		
		0b	Tiling Di	sabled (Linear	Blit)			
		1b	Tiling Er	nabled			Tile-X or Tile-Y.	
	10:8	Reserved						
		Format:			٩	MBZ		
	7:0	DWord Length						
		Default Val	lue:				06h	
1	31	Reserved						
		Format: MBZ						
BR01	30	Clipping Er	nabled					
			Value				Name	
		0b			Disabled			



		XY_SE	TUP_BLT			
		1b	Enabled			
	29	Mono Source Transparency Mode				
		Value	Name			
		0b U	se Background			
		1b Transparency Enabled				
	28:26	Reserved				
		Format:	MBZ			
	25:24	Color Depth				
		Value	Name			
		00b	8 Bit Color			
		01b	16 Bit Color(565)			
		10b	16 Bit Color(1555)			
		11b	32 Bit Color			
	23:16	Raster Operation				
	15:0	Destination Pitch in	DWords			
		2's complement (Negative Pitch Not allowed for Pixel nor Text) For Tiled surfaces (bit_11 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).				
2	31:16	ClipRect Y1 Coordinate (Top) (30:16 = 15 bit positive number)				
BR24	15:0	ClipRect X1 Coordinate (Left) (14:00 = 15 bit positive number)				
3	31:16	ClipRect Y2 Coordina (30:16 = 15 bit positiv	a te (Bottom) e number)			
BR25	15:0	ClipRect X2 Coordina (14:00 = 15 bit positiv	e number)			
4	31:0	Setup Destination Ba	ase Address			
		Format:	GraphicsAddress[31:0]			
BR09		Base address of the de enabled), this address	estination surface: X=0, Y=0. When Tiling is enabled (Bit_11 is limited to 4Kbytes.			
5	31:0	Setup Background C	olor 15:01 32 bit - [31:0] All			
BR05			13.0], 32 bit - [31.0] All			
6 BR06	31:0	Setup Foreground Co 8 bit = [7:0], 16 bit = [olor [15:0], 32 bit = [31:0] (SLB and TB only)			
7	31.0	Setup Pattern Base A	Address for Color Pattern			
,	51.0	Format:	GraphicsAddress[31:0]			
BR07		(26:06 are implemente data must be located	ed) (SLB only) (Note no NPO2 change here). The pattern in linear memory.			



BlitterCS

2

XY_SETUP_CLIP_BLT

Source:

Length Bias:

This command is used to only change the clip coordinate registers. These are the same clipping registers as the Setup clipping registers above.

DWord	Bit	Description				
0	31:29	Client				
		Default Valu	ie:	02h 2D Pro	ocessor	
BR00		Format:		Opcode		
	28:22	Instruction	Target(Opcode)	-		
		Default Valu	ie:		03h	
		Format:			Opcode	2
	21:12	Reserved				
		Format:			MBZ	
	11	Tiling Enabl	e			
		Value Name				
		0b	Tiling Disabled (Linear Blit)			
		1b	Tiling Enabled (Tile-X or Tile-Y			
	10:8	Reserved	Reserved			
		Format:			MBZ	
	7:0	DWord Leng	gth			
		Default Valu	ie:			01h
1	31:16	ClipRect Y1	Coordinate (Top)		
5524		(30:16 = 15	oit positive numbe	er)		
BR24	15:0	ClipRect X1	Coordinate (Left	t)		
2	21.1.6	(14:00 = 15 bit positive number)				
2	31:16	ClipRect Y2 Coordinate (Bottom)				
BR25	15:0	ClipRect X2	Coordinate (Rig	ht)		
		(14:00 = 15	pit positive number	er)		



		XY_SE1	UP_		_ P	ATTERN	I_SL_BL	.т
Source:		BlitterCS						
Length Bi	ias:	2						
This setup instruction supplies common setup information including clipping coordinates used exclusivel the following instruction: XY_SCANLINE_BLT (SLB) - 1 scan line of monochrome pattern and destination a only operands allowed.					s used exclusively with and destination are the			
DWord	Bit				D	Description		
0	31:29	Client						
		Default Value: 02h 2D Processor						
BR00		Format:				Opcode		
	28:22	Instruction Targe	et(Opc	ode)				
		Default Value:	· •				11h	
		Format:					Opcode	
	21:20	32 bpp Byte Mas	k					
		Value				Ν	lame	
		1xb		Write Alpha C	hanr	nel		
x1b Write RGB Channel								
	19:12	Reserved						
		Format:					MBZ	
	11	Tiling Enable	-					
		Value				Nan	ne	
		0b	Tiling	Disabled (Line	ear B	lit)		
		1b	Tiling	Enabled (Tile-	X or	Tile-Y		
	10:8	Reserved						
		Format:					MBZ	
	7:0	DWord Length						11
		Default Value:						07h
1	31	Solid Pattern Sel (SLB and Pixel onl	ect y)					
BR01		Value					Name	
		0		No Solid F	Patte	rn		
		1		Solid Patte	ern			
	30	Clipping Enabled						
			Value				Name	9
		0b			D	isabled		
		1b			Er	nabled		
	29	Reserved				r		
Format: MBZ								



		XY_SETUP	_MONO_PATTERN_SL_BLT					
	28	Mono Pattern Transpar	ency Mode					
		Value	Name					
		Ob L	Jse Background					
		1b T	ransparency Enabled					
	27:26	Reserved						
		Format:	Format: MBZ					
	25:24	Color Depth						
		Value	Name					
		00b	8 Bit Color					
		01b	16 Bit Color(565)					
		10b	16 Bit Color(1555)					
		11b	32 Bit Color					
	23:16	Raster Operation						
	15:0	Destination Pitch in DWords 2's complement (Negative Pitch Not allowed for Pixel nor Text) For Tiled surfaces (bit_11 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).						
2	31:16	ClipRect Y1 Coordinate (Top) (30:16 = 15 bit positive number)						
BR24	15:0	ClipRect X1 Coordinate (Left) (14:00 = 15 bit positive number)						
3	31:16	ClipRect Y2 Coordinate (30:16 = 15 bit positive n	(Bottom) umber)					
BR25	15:0	ClipRect X2 Coordinate (14:00 = 15 bit positive n	(Right) umber)					
4	31:0	Setup Destination Base	Address					
DD OO		Format:	GraphicsAddress[31:0]					
вкоя		Base address of the desti address is limited to 4Kb	nation surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this ytes.					
5	31:0	Setup Background Colo	r					
BBOE		8 bit = [7:0], 16 bit = [15:	0], 32 bit = [31:0] All					
BRUS	21.0	Cature Faus manual Cala	_					
б	31:0	Setup Foreground Colo 8 bit = $[7:0]$ 16 bit = $[15:$	r 01 32 bit = $[31:0]$ (SLB and TB only)					
BR06								
7	31:0	DW0 (least significant)	for a Monochrome Pattern					
BR20								
8 BR21	31:0	DW1 (most significant)	for a Monochrome Pattern					



XY	SRC	COPY	BLT
_		-	

Source: BlitterCS

Length Bias:

This BLT instruction performs a color source copy where the only operands involved is a color source and destination of the same bit width.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is less than Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is less than Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

The ROP value chosen must involve source and no pattern data in the ROP operation.

DWord	Bit	Description							
0	31:29	Client							
		Default Value:			02h 2D Processo	r			
BR00		Format:	nat: Opcode						
	28:22	Instruction Ta	rget(Opc	ode)					
		Default Value:				53h			
		Format:		Opcode					
	21:20	32bpp Byte Mask This field is only used for 32bpp.							
		Value	•	Name					
		00b		[Default]					
		1xb		Write Alpha Chanr	nel				
		x1b		Write RGB Channe					
	19:16	Reserved							
		Format:			M	BZ			
	15	Src Tiling Ena	ble						
		Value		Name		Description			
		0b Tiling Disabled (Linear)							
		1bTiling EnabledTile-X or Tile-Y.							
	14:12	Reserved							
		Format:			M	BZ			

		XY_S	ETUP_	MONO_	PA ⁻	FTERI	N_SL	BLT
	11	Dest Tiling E	nable					
		Value		Nan	ne			Description
		0b	Tiling Disabl	ed (Linear Blit))			
		1b Tiling Enabled						Tile-X or Tile-Y.
	10:8	Reserved	Reserved					
		Format:					MBZ	
	7:0 DWord Length							
			Value					Name
		06h						
1	31	Reserved						
BR13		Format:					MBZ	
	30	Clipping Ena	bled					News
		Value			Dicah			Name
		00 1b				Enabled		
	20.26	Pecentred			спарт	20		
	29.20	Format [.]					MB7	
	25.24	24 Color Depth						
	23.21	Val	ue				Name	
		00b		8 Bit Color				
		01b	16 Bit Color(565)					
		10b		16 Bit Color(1	.555)			
		11b		32 Bit Color				
	23:16	Raster Opera	tion					
	15:0	Destination I	Pitch in DWo	ords				
		2's compleme	ent For Tiled s rity for Tile-V	surfaces (bit_1)	1 enab	led) this p	itch is of	512Byte granularity for Tile-X,
2	31:16	Destination	1 Coordinat	te (Top)	10 120	indytes (of		
		16 bit signed	number.					
BR22	15:0	Destination X	K1 Coordina t	te (Left)				
		16 bit signed	number.					
3	31:16	Destination	/2 Coordina t	te (Bottom)				
BR23	15:0	Destination 2	K2 Coordinat	te (Riaht)				
		16 bit signed	number.	··· (··· ·· ··				
4	31:0	Destination I	Base Address	5				
BROQ		Format:		GraphicsAddr	ress[31	:0]		
DRUY		Base address	of the destin	ation surface: 2	X=0, Y	=0. When	Dest Tili	ng is enabled (Bit_11 enabled),
this address is limited to 4Kbytes.								

this address is limited to 4Kbytes.



XY_SETUP_MONO_PATTERN_SL_BLT 5 31:16 **Source Y1 Coordinate (Top)** 16 bit signed number. BR26 15:0 Source X1 Coordinate (Left) 16 bit signed number. 6 31:16 **Reserved** Format: MBZ BR11 15:0 Source Pitch (double word aligned) and in DWords 2's complement. For Tiled Src (bit 15 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Yand can be upto 128Kbytes (or 32KDwords). 7 Source Base Address 31:0 Format: GraphicsAddress[31:0]

Base address of the destination surface: X=0, Y=0. When Src Tiling is enabled (Bit_15 enabled),

BR12



XY_SRC_COPY_CHROMA_BLT

Source: BlitterCS

2

Length Bias:

This BLT instruction performs a color source copy with chroma-keying where the only operands involved is a color source and destination of the same bit width.

The source and destination operands may overlap, which means that the X and Y directions can be either forward or backwards. The BLT Engine takes care of all situations. The base addresses plus the X and Y coordinates determine if there is an overlap between the source and destination operands. If the base addresses of the source and destination are the same and the Source X1 is less than Destination X1, then the BLT Engine performs the accesses in the X-backwards access pattern. There is no need to look for an actual overlap. If the base addresses are the same and Source Y1 is less than Destination Y1, then the scan line accesses start at Destination Y2 with the corresponding source scan line and the strides are subtracted for every scan line access.

The ROP value chosen must involve source and no pattern data in the ROP operation.

DWord	Bit	Description						
0	31:29	Client						
		Default Value: 02h 2D F				2D Processor		
BR00		Format:			Opcode			
	28:22	Instruction Ta	Instruction Target(Opcode)					
		Default Value	:			73	h	
		Format:				Op	ocode	
	21:20	32bpp Byte N	lask					
		This field is only used for 32bpp.						
Value Name					Name		•	
00b [Default]								
		1xb		Write Alpha Chani	nel			
		x1b		Write RGB Channe	el			
	19:17	7 Transparency Range Mode (chroma-key)						
	16	Reserved						
		Format:				MBZ	,	
	15	Src Tiling Enable						
		Value		Name			Description	
		0b	Tiling Di	sabled (Linear)				
		1b	Tiling Enabled			Tile-X or Tile-Y.		
	14:12	Reserved						
		Format:		MBZ				



		>	Y_SRC_	COPY_C	CHR		BL1	г	
	11	Dest Tiling	Dest Tiling Enable						
		Value		Nar	ne			Description	
		0b	Tiling Disab	Tiling Disabled (Linear Blit)					
		1b	Tiling Enabl	ed				Tile-X or Tile-Y.	
	10:8	Reserved							
		Format:					MBZ		
	7:0	DWord Len	DWord Length						
			Value	•				Name	
		08h							
1	31	Reserved							
BR13		Format:					MBZ		
220	30	Clipping En	abled		1				
			Value					Name	
		00			Disab	led	1		
		10			Enabl	ed			
29:26 Reserved									
	05.04	MBZ							
	25:24	Value							
		00b	aiue	8 Bit Color			Name		
		005 01b	16 Bit Color(565)						
		10b	16 Bit Color(1555)						
		11b		32 Bit Color					
	23:16	Raster Ope	ration						
	15:0	Destination	Pitch in DW	ords					
		2's complen	nent For Tiled	surfaces (bit_1	L1 enab	oled) this _l	oitch is o	f 512Byte granularity for Tile-	
		X, 128B grar	nularity for Tile	e-Y and can be	e upto	128Kbyte	s (or 32K	Dwords).	
2	31:16	Destination 16 bit signe	Y1 Coordina d number.	ite (Top)					
BR22	15:0	Destination 16 bit signe	X1 Coordina d number.	ite (Left)					
3	31:16	Destination 16 bit signe	Y2 Coordina d number.	ite (Bottom)					
BR23	15:0	Destination	X2 Coordina	te (Right)					
4	31:0	Destination	Base Addres	S					
		Format:		GraphicsAdd	ress[31	.:0]			
BR09 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (this address is limited to 4Kbytes.				s enabled (Bit_11 enabled),					



XY_SRC_COPY_CHROMA_BLT

5	31:16	Source Y1 Coordinate (Top) 16 bit signed number.					
BR26	15:0	Source X1 Coordinate (L 16 bit signed number.	eft)				
6	31:16	Reserved					
0011		Format:		MBZ			
BR11	15:0	Source Pitch (double word aligned) and in DWords 2's complement. For Tiled Src (bit 15 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Yand can be upto 128Kbytes (or 32KDwords).					
7	31:0	Source Base Address					
		Format:	GraphicsAddress[31:0]				
BR12		Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_15 enabled), this address is limited to 4Kbytes.					
8	31:0	Transparency Color Low					
BR18		(Chroma-key Low = Pixel Greater or Equal)					
9 BR19	31:0	Transparency Color High (Chroma-key High = Pixel	Less or Equal)				



XY	TEXT	BLT
-		_

Source: BlitterCS

Length Bias:

All source scan lines and pixels that fall within the ClipRect Y and X coordinates are written. The source address corresponds to Destination X1 and Y1 coordinate.

Text is either bit or byte packed. Bit packed means that the next scan line starts 1 pixel after the end of the current scan line with no bit padding. Byte packed means that the next scan line starts on the first bit of the next byte boundary after the last bit of the current line.

Source expansion color registers are always in the SETUP_BLT.

Negative Stride (= Pitch) is NOT ALLOWED.

DWord	Bit	Description						
0	31:29	Client						
		Default Value: 02h 2D Pro				cessor		
BR00		Format:		Ор	code			
	28:22	Instruction	Target(Opcode)					
		Default Val	ue:			26h		
		Format:				Орсо	de	
	21:17	Reserved						
		Format:			١	MBZ		
	16	Bit / Byte P	acked		·			
		Byte packed	is for the NT driver.					
		Value Nam					ne	
		0			Bit			
		1 Byte						
	15:12	Reserved						
		Format:			١	MBZ		
	11	Tiling Enab	le					
		Value	Name					Description
		0b	Tiling Disabled (Linear Blit)					
		1b	Tiling Enabled				Tile-X	or Tile-Y.
	10:8	Reserved						
		Format: MBZ						
	7:0	DWord Len	gth					
		Default Val	ue:					02h
1	31:16	Destination	Y1 Coordinate (Top)					



	XY_TEXT_BLT							
		16 bit signed number.	16 bit signed number.					
BR22 15:0 Destination X1 Coordinate (Left) 16 bit signed number.								
2	31:16	Destination Y2 Coordin 16 bit signed number.	Destination Y2 Coordinate (Bottom) 16 bit signed number.					
BR23	15:0	Destination X2 Coordin 16 bit signed number.	nate (Right)					
3	31:0	Source Address						
		Format:	GraphicsAddress[31:0]					
BR12		(address of the first byte on scan line corresponding to Dst X1,Y1). (Note no NPO2 change here)						



XY_TEXT_IMMEDIATE_BLT

Source: BlitterCS

Length Bias:

This instruction allows the Driver to send data through the instruction stream that eliminates the read latency of reading a source from memory.

If an operand is in system cacheable memory and either small or only accessed once, it can be copied directly to the instruction stream versus to graphics accessible memory. The IMMEDIATE_BLT data MUST transfer an even number of doublewords.

The BLT engine will hang if it does not get an even number of doublewords. All source scan lines and pixels that fall within the ClipRect X and Y coordinates are written. The source data corresponds to Destination X1 and Y1 coordinate.

Source expansion color registers are always in the SETUP_BLT. NEGATIVE STRIDE (= PITCH) IS NOT ALLOWED.

DWord	Bit	Description						
0	31:29	Client						
		Default Value: 02h 2D Pr			2D Proce	ocessor		
BR00		Format: Opcode			code			
	28:22	Instruction	Target(Opcode)					
		Default Val	ue:	31h				
		Format:				Орс	ode	
21:17 Reserved								
		Format: MBZ						
	16 Bit / Byte Packed Byte packed is for the NT driver.							
			Value				Name	
		0			Bit			
		1			Byte			
	15:12	Reserved						
		Format:				MBZ		
	11	Tiling Enable						
		Value	Name				Description	
		0b	Tiling Disabled (Linear Blit)					
		1b Tiling Enabled Tile-X				Tile-X or Tile-Y.		
	10:8	Reserved						
Format: MBZ								



XY_TEXT_IMMEDIATE_BLT				
	7:0	DWord Length		
		Default Value:	01h Excludes DWORD 0,1	
		01 + DWL = (Number of Immediate double words)h		
1	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.		
BR22	15:0	Destination X1 Coordinate (Left) 16 bit signed number.		
2	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.		
BR23	15:0	Destination X2 Coordinate (Right) 16 bit signed number.		
3n	31:0	Immediate Data		