



Intel® Open Source HD Graphics and Intel Iris™ Plus Graphics

Programmer's Reference Manual

For the 2016 - 2017 Intel Core™ Processors, Celeron™ Processors,
and Pentium™ Processors based on the "Kaby Lake" Platform

Volume 2a: Command Reference: Instructions (Command Opcodes)

January 2017, Revision 1.0



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 © 2017, Intel Corporation. All rights reserved.



Table of Contents

3DPRIMITIVE	1
3DSTATE_AA_LINE_PARAMETERS	6
3DSTATE_BINDING_TABLE_EDIT_DS	8
3DSTATE_BINDING_TABLE_EDIT_GS	10
3DSTATE_BINDING_TABLE_EDIT_HS	12
3DSTATE_BINDING_TABLE_EDIT_PS	14
3DSTATE_BINDING_TABLE_EDIT_VS	16
3DSTATE_BINDING_TABLE_POINTERS_DS	18
3DSTATE_BINDING_TABLE_POINTERS_GS	20
3DSTATE_BINDING_TABLE_POINTERS_HS	22
3DSTATE_BINDING_TABLE_POINTERS_PS	24
3DSTATE_BINDING_TABLE_POINTERS_VS	26
3DSTATE_BINDING_TABLE_POOL_ALLOC	28
3DSTATE_BLEND_STATE_POINTERS	30
3DSTATE_CC_STATE_POINTERS	31
3DSTATE_CHROMA_KEY	32
3DSTATE_CLEAR_PARAMS	34
3DSTATE_CLIP	36
3DSTATE_CONSTANT_DS	42
3DSTATE_CONSTANT_GS	44
3DSTATE_CONSTANT_HS	46
3DSTATE_CONSTANT_PS	48
3DSTATE_CONSTANT_VS	50
3DSTATE_DEPTH_BUFFER	52
3DSTATE_DRAWING_RECTANGLE	61
3DSTATE_DS	64
3DSTATE_DX9_CONSTANT_BUFFER_POOL_ALLOC	72
3DSTATE_DX9_CONSTANTB_PS	74
3DSTATE_DX9_CONSTANTB_VS	76
3DSTATE_DX9_CONSTANTF_PS	78
3DSTATE_DX9_CONSTANTF_VS	80



3DSTATE_DX9_CONSTANTI_PS.....	82
3DSTATE_DX9_CONSTANTI_VS	84
3DSTATE_DX9_GENERATE_ACTIVE_PS.....	86
3DSTATE_DX9_GENERATE_ACTIVE_VS	89
3DSTATE_DX9_LOCAL_VALID_PS.....	92
3DSTATE_DX9_LOCAL_VALID_VS.....	93
3DSTATE_GATHER_CONSTANT_DS.....	94
3DSTATE_GATHER_CONSTANT_GS.....	97
3DSTATE_GATHER_CONSTANT_HS.....	100
3DSTATE_GATHER_CONSTANT_PS	103
3DSTATE_GATHER_CONSTANT_VS	107
3DSTATE_GATHER_POOL_ALLOC	111
3DSTATE_GS	113
3DSTATE_HIER_DEPTH_BUFFER.....	124
3DSTATE_HS	127
3DSTATE_INDEX_BUFFER.....	134
3DSTATE_LINE_STIPPLE	136
3DSTATE_MONOFILTER_SIZE	138
3DSTATE_MULTISAMPLE.....	140
3DSTATE_POLY_STIPPLE_OFFSET	142
3DSTATE_POLY_STIPPLE_PATTERN.....	144
3DSTATE_PS_BLEND	145
3DSTATE_PS.....	147
3DSTATE_PS_EXTRA	154
3DSTATE_PUSH_CONSTANT_ALLOC_DS	158
3DSTATE_PUSH_CONSTANT_ALLOC_GS	160
3DSTATE_PUSH_CONSTANT_ALLOC_HS	162
3DSTATE_PUSH_CONSTANT_ALLOC_PS.....	164
3DSTATE_PUSH_CONSTANT_ALLOC_VS.....	166
3DSTATE_RASTER	168
3DSTATE_RS_CONSTANT_POINTER.....	173
3DSTATE_SAMPLE_MASK	175
3DSTATE_SAMPLE_PATTERN	177



3DSTATE_SAMPLER_PALETTE_LOAD0	187
3DSTATE_SAMPLER_PALETTE_LOAD1	188
3DSTATE_SAMPLER_STATE_POINTERS_DS	190
3DSTATE_SAMPLER_STATE_POINTERS_GS	191
3DSTATE_SAMPLER_STATE_POINTERS_HS	192
3DSTATE_SAMPLER_STATE_POINTERS_PS	193
3DSTATE_SAMPLER_STATE_POINTERS_VS	194
3DSTATE_SBE	195
3DSTATE_SBE_SWIZ	203
3DSTATE_SCISSOR_STATE_POINTERS	205
3DSTATE_SF	206
3DSTATE_SO_BUFFER	211
3DSTATE_SO_DECL_LIST	214
3DSTATE_STENCIL_BUFFER	217
3DSTATE_STREAMOUT	220
3DSTATE_TE	225
3DSTATE_URB_CLEAR	229
3DSTATE_URB_DS	231
3DSTATE_URB_GS	233
3DSTATE_URB_HS	235
3DSTATE_URB_VS	237
3DSTATE_VERTEX_BUFFERS	239
3DSTATE_VERTEX_ELEMENTS	241
3DSTATE_VF_COMPONENT_PACKING	243
3DSTATE_VF	247
3DSTATE_VF_INSTANCING	249
3DSTATE_VF_SGVS	251
3DSTATE_VF_STATISTICS	254
3DSTATE_VF_TOPOLOGY	255
3DSTATE_VIEWPORT_STATE_POINTERS_CC	256
3DSTATE_VIEWPORT_STATE_POINTERS_SF_CLIP	257
3DSTATE_VS	258
3DSTATE_WM_CHROMAKEY	266



3DSTATE_WM_DEPTH_STENCIL..... 267

3DSTATE_WM..... 271

3DSTATE_WM_HZ_OP 277

A64 Byte Scaled Read MSD 283

A64 Byte Scaled Write MSD 285

A64 Byte Scattered Read MSD..... 287

A64 Byte Scattered Write MSD..... 288

A64 Dword Scaled Read MSD 289

A64 Dword Scaled Write MSD 291

A64 Dword Scattered Read MSD..... 293

A64 Dword Scattered Write MSD..... 294

A64 Dword SIMD4x2 Untyped Atomic Float Binary with Return Data Operation MSD 295

A64 Dword SIMD4x2 Untyped Atomic Float Binary Write Only Operation MSD 297

A64 Dword SIMD4x2 Untyped Atomic Float Ternary with Return Data Operation MSD 299

A64 Dword SIMD4x2 Untyped Atomic Float Ternary Write Only Operation MSD 301

A64 Dword SIMD4x2 Untyped Atomic Integer Binary with Return Data Operation MSD303

A64 Dword SIMD4x2 Untyped Atomic Integer Binary Write Only Operation MSD 304

A64 Dword SIMD4x2 Untyped Atomic Integer Ternary with Return Data Operation MSD305

A64 Dword SIMD4x2 Untyped Atomic Integer Ternary Write Only Operation MSD 306

A64 Dword SIMD4x2 Untyped Atomic Integer Unary with Return Data Operation MSD307

A64 Dword SIMD4x2 Untyped Atomic Integer Unary Write Only Operation MSD 308

A64 Dword Untyped Atomic Float Binary with Return Data Operation MSD 309

A64 Dword Untyped Atomic Float Binary Write Only Operation MSD..... 311

A64 Dword Untyped Atomic Float Ternary with Return Data Operation MSD 313

A64 Dword Untyped Atomic Float Ternary Write Only Operation MSD 315

A64 Dword Untyped Atomic Integer Binary with Return Data Operation MSD 317

A64 Dword Untyped Atomic Integer Binary Write Only Operation MSD..... 318

A64 Dword Untyped Atomic Integer Ternary with Return Data Operation MSD 319

A64 Dword Untyped Atomic Integer Ternary Write Only Operation MSD 320

A64 Dword Untyped Atomic Integer Unary with Return Data Operation MSD 321

A64 Dword Untyped Atomic Integer Unary Write Only Operation MSD 322

A64 Hword Block Read MSD..... 323

A64 Hword Block Write MSD..... 324



A64 Oword Block Read MSD	325
A64 Oword Block Write MSD	326
A64 Oword Dual Block Read MSD	327
A64 Oword Dual Block Write MSD	328
A64 Oword Unaligned Block Read MSD	329
A64 Qword Scaled Read MSD	330
A64 Qword Scaled Write MSD	332
A64 Qword Scattered Read MSD	334
A64 Qword Scattered Write MSD	335
A64 Qword SIMD4x2 Untyped Atomic Integer Binary with Return Data Operation MSD 336	
A64 Qword SIMD4x2 Untyped Atomic Integer Binary Write Only Operation MSD	337
A64 Qword SIMD4x2 Untyped Atomic Integer Trinary with Return Data Operation MSD 338	
A64 Qword SIMD4x2 Untyped Atomic Integer Trinary Write Only Operation MSD	339
A64 Qword SIMD4x2 Untyped Atomic Integer Unary with Return Data Operation MSD 340	
A64 Qword SIMD4x2 Untyped Atomic Integer Unary Write Only Operation MSD	341
A64 Qword Untyped Atomic Integer Binary with Return Data Operation MSD	342
A64 Qword Untyped Atomic Integer Binary Write Only Operation MSD	343
A64 Qword Untyped Atomic Integer Trinary with Return Data Operation MSD	344
A64 Qword Untyped Atomic Integer Trinary Write Only Operation MSD	345
A64 Qword Untyped Atomic Integer Unary with Return Data Operation MSD	346
A64 Qword Untyped Atomic Integer Unary Write Only Operation MSD	347
A64 Scaled Untyped Surface Read MSD	348
A64 Scaled Untyped Surface Write MSD	349
A64 Untyped Surface Read MSD	350
A64 Untyped Surface Write MSD	351
Addition	352
Addition with Carry	354
Arithmetic Shift Right	355
Average	357
Bit Field Extract	358
Bit Field Insert 1	362
Bit Field Insert 2	364
Bit Field Reverse	368



Branch Converging	369
Branch Diverging	371
Break	373
Byte Scaled Read MSD	374
Byte Scaled Write MSD	375
Byte Scattered Read MSD	376
Byte Scattered Write MSD	378
Call	380
Call Absolute	382
Compare	384
Compare NaN	386
Conditional Select	388
Conditional Send Message	391
Conditional Split Send Message	393
Constant Cache Dword Scattered Read MSD	396
Constant Cache Oword Block Read MSD	398
Constant Cache Oword Dual Block Read MSD	399
Constant Cache Oword Unaligned Block Read MSD	400
Continue	401
Count Bits Set	403
Dot Product 2	404
Dot Product 3	406
Dot Product 4	408
Dot Product Homogeneous	410
Dword Atomic Counter Binary with Return Data Operation MSD	412
Dword Atomic Counter Binary Write Only Operation MSD	413
Dword Atomic Counter Unary with Return Data Operation MSD	414
Dword Atomic Counter Unary Write Only Operation MSD	415
Dword Scattered Read MSD	416
Dword Scattered Write MSD	418
Dword SIMD4x2 Atomic Counter Binary with Return Data Operation MSD	420
Dword SIMD4x2 Atomic Counter Binary Write Only Operation MSD	421
Dword SIMD4x2 Atomic Counter Unary with Return Data Operation MSD	422



Dword SIMD4x2 Atomic Counter Unary Write Only Operation MSD 423

Dword SIMD4x2 Typed Atomic Integer Binary with Return Data Operation MSD..... 424

Dword SIMD4x2 Typed Atomic Integer Binary Write Only Operation MSD..... 425

Dword SIMD4x2 Typed Atomic Integer Ternary with Return Data Operation MSD 426

Dword SIMD4x2 Typed Atomic Integer Ternary Write Only Operation MSD 427

Dword SIMD4x2 Typed Atomic Integer Unary with Return Data Operation MSD 428

Dword SIMD4x2 Typed Atomic Integer Unary Write Only Operation MSD 429

Dword SIMD4x2 Untyped Atomic Float Binary with Return Data Operation MSD 430

Dword SIMD4x2 Untyped Atomic Float Binary Write Only Operation MSD 432

Dword SIMD4x2 Untyped Atomic Float Ternary with Return Data Operation MSD..... 434

Dword SIMD4x2 Untyped Atomic Float Ternary Write Only Operation MSD 436

Dword SIMD4x2 Untyped Atomic Integer Binary with Return Data Operation MSD 438

Dword SIMD4x2 Untyped Atomic Integer Binary Write Only Operation MSD 439

Dword SIMD4x2 Untyped Atomic Integer Ternary with Return Data Operation MSD ... 440

Dword SIMD4x2 Untyped Atomic Integer Ternary Write Only Operation MSD 441

Dword SIMD4x2 Untyped Atomic Integer Unary with Return Data Operation MSD 442

Dword SIMD4x2 Untyped Atomic Integer Unary Write Only Operation MSD 443

Dword Typed Atomic Integer Binary with Return Data Operation MSD..... 444

Dword Typed Atomic Integer Binary Write Only Operation MSD 445

Dword Typed Atomic Integer Ternary with Return Data Operation MSD 446

Dword Typed Atomic Integer Ternary Write Only Operation MSD..... 447

Dword Typed Atomic Integer Unary with Return Data Operation MSD..... 448

Dword Typed Atomic Integer Unary Write Only Operation MSD..... 449

Dword Untyped Atomic Float Binary with Return Data Operation MSD 450

Dword Untyped Atomic Float Binary Write Only Operation MSD..... 452

Dword Untyped Atomic Float Ternary with Return Data Operation MSD 454

Dword Untyped Atomic Float Ternary Write Only Operation MSD 456

Dword Untyped Atomic Integer Binary with Return Data Operation MSD 458

Dword Untyped Atomic Integer Binary Write Only Operation MSD..... 459

Dword Untyped Atomic Integer Ternary with Return Data Operation MSD 460

Dword Untyped Atomic Integer Ternary Write Only Operation MSD 461

Dword Untyped Atomic Integer Unary with Return Data Operation MSD 462

Dword Untyped Atomic Integer Unary Write Only Operation MSD 463



Else	464
End If.....	466
Extended Math Function	468
Find First Bit from LSB Side.....	472
Find First Bit from MSB Side	474
Fraction.....	476
Goto	477
GPGPU_CSR_BASE_ADDRESS	479
GPGPU_WALKER	481
Half Precision HI8DS Render Target Write MSD	485
Half Precision LO8DS Render Target Write MSD	488
Half Precision REP16 Render Target Write MSD	491
Half Precision SIMD8 Render Target Write MSD	494
Half Precision SIMD16 Render Target Write MSD	497
Halt.....	500
HCP_BSD_OBJECT	502
HCP_FQM_STATE.....	504
HCP_IND_OBJ_BASE_ADDR_STATE	507
HCP_PAK_INSERT_OBJECT	510
HCP_PAK_OBJECT	514
HCP_PIC_STATE	516
HCP_PIPE_BUF_ADDR_STATE	532
HCP_PIPE_MODE_SELECT	538
HCP_QM_STATE.....	542
HCP_REF_IDX_STATE.....	545
HCP_SLICE_STATE	548
HCP_SURFACE_STATE.....	559
HCP_TILE_STATE.....	562
HCP_VP9_PAK_OBJECT.....	566
HCP_VP9_PIC_STATE.....	568
HCP_VP9_SEGMENT_STATE.....	583
HCP_WEIGHTOFFSET_STATE	587
HEVC_VP9_RDOQ_STATE	590



HI8DS Render Target Write MSD	594
HUC_CFG_STATE	597
HUC_DMEM_STATE	599
HUC_IMEM_STATE	601
HUC_IND_OBJ_BASE_ADDR_STATE	603
HUC_PIPE_MODE_SELECT	605
HUC_START	607
HUC_STREAM_OBJECT	608
HUC_VIRTUAL_ADDR_STATE	611
If	612
Illegal	614
Integer Subtraction with Borrow	615
Join	616
Jump Indexed	618
Leading Zero Detection	620
Line	621
Linear Interpolation	623
LO8DS Render Target Write MSD	626
Logic And	629
Logic Not	631
Logic Or	633
Logic Xor	635
MEDIA_CURBE_LOAD	637
MEDIA_INTERFACE_DESCRIPTOR_LOAD	639
MEDIA_OBJECT	641
MEDIA_OBJECT_GRPID	645
MEDIA_OBJECT_PRT	649
MEDIA_OBJECT_WALKER	651
MEDIA_STATE_FLUSH	658
MEDIA_VFE_STATE	660
Media Block Read MSD	667
Media Block Write MSD	668
Media Transpose Read MSD	669



Memory Fence MSD	670
MFC_AVC_PAK_OBJECT	672
MFC_JPEG_HUFF_TABLE_STATE	674
MFC_JPEG_SCAN_OBJECT	676
MFC_MPEG2_PAK_OBJECT	679
MFC_MPEG2_SLICEGROUP_STATE	681
MFD_AVC_BSD_OBJECT	689
MFD_AVC_DPB_STATE	692
MFD_AVC_PICID_STATE	695
MFD_AVC_SLICEADDR	697
MFD_IT_OBJECT	700
MFD_JPEG_BSD_OBJECT	703
MFD_MPEG2_BSD_OBJECT	706
MFD_VC1_BSD_OBJECT	708
MFD_VC1_LONG_PIC_STATE	711
MFD_VC1_SHORT_PIC_STATE	724
MFD_VP8_BSD_OBJECT	733
MFV_AVC_DIRECTMODE_STATE	740
MFV_AVC_IMG_STATE	745
MFV_AVC_REF_IDX_STATE	765
MFV_AVC_SLICE_STATE	768
MFV_AVC_WEIGHTOFFSET_STATE	782
MFV_BSP_BUF_BASE_ADDR_STATE	785
MFV_DBK_OBJECT	791
MFV_FQM_STATE	798
MFV_IND_OBJ_BASE_ADDR_STATE	800
MFV_JPEG_HUFF_TABLE_STATE	812
MFV_JPEG_PIC_STATE	814
MFV_MPEG2_PIC_STATE	821
MFV_PAK_INSERT_OBJECT	835
MFV_PIPE_BUF_ADDR_STATE	839
MFV_PIPE_MODE_SELECT	864
MFV_QM_STATE	870



MFX_STATE_POINTER	872
MFX_STITCH_OBJECT	874
MFX_SURFACE_STATE	876
MFX_VC1_DIRECTMODE_STATE	884
MFX_VC1_PRED_PIPE_STATE	888
MFX_VP8_BSP_BUF_BASE_ADDR_STATE	894
MFX_VP8_Encoder_CFG	908
MFX_VP8_PAK_OBJECT	921
MFX_VP8_PIC_STATE	923
MFX_WAIT	952
MI_ARB_CHECK	953
MI_ARB_CHECK	954
MI_ARB_CHECK	955
MI_ARB_CHECK	956
MI_ARB_ON_OFF	957
MI_ATOMIC	959
MI_BATCH_BUFFER_END	964
MI_BATCH_BUFFER_END	965
MI_BATCH_BUFFER_END	966
MI_BATCH_BUFFER_END	967
MI_BATCH_BUFFER_START	968
MI_CLFLUSH	972
MI_CONDITIONAL_BATCH_BUFFER_END	974
MI_COPY_MEM_MEM	977
MI_COPY_MEM_MEM	979
MI_COPY_MEM_MEM	981
MI_COPY_MEM_MEM	983
MI_DISPLAY_FLIP	985
MI_FLUSH_DW	990
MI_FLUSH_DW	993
MI_FLUSH_DW	996
MI_FORCE_WAKEUP	999
MI_LOAD_REGISTER_IMM	1001



MI_LOAD_REGISTER_MEM	1003
MI_LOAD_REGISTER_REG	1005
MI_LOAD_SCAN_LINES_EXCL	1007
MI_LOAD_SCAN_LINES_EXCL	1009
MI_LOAD_SCAN_LINES_INCL	1011
MI_LOAD_SCAN_LINES_INCL	1013
MI_LOAD_URB_MEM	1015
MI_MATH	1016
MI_MATH	1017
MI_MATH	1018
MI_MATH	1019
MI_NOOP	1020
MI_NOOP	1021
MI_NOOP	1022
MI_NOOP	1023
MI_PREDICATE	1024
MI_REPORT_HEAD	1026
MI_REPORT_HEAD	1027
MI_REPORT_HEAD	1028
MI_REPORT_HEAD	1029
MI_REPORT_PERF_COUNT	1030
MI_RS_CONTEXT	1032
MI_RS_CONTROL	1033
MI_RS_STORE_DATA_IMM	1035
MI_SEMAPHORE_SIGNAL	1037
MI_SEMAPHORE_WAIT	1040
MI_SET_CONTEXT	1043
MI_SET_PREDICATE	1046
MI_STORE_DATA_IMM	1048
MI_STORE_DATA_INDEX	1051
MI_STORE_DATA_INDEX	1053
MI_STORE_DATA_INDEX	1055
MI_STORE_DATA_INDEX	1057



MI_STORE_REGISTER_MEM	1059
MI_STORE_URB_MEM	1062
MI_SUSPEND_FLUSH	1064
MI_SUSPEND_FLUSH	1065
MI_SUSPEND_FLUSH	1066
MI_SUSPEND_FLUSH	1067
MI_TOPOLOGY_FILTER	1068
MI_UPDATE_GTT	1069
MI_URB_ATOMIC_ALLOC	1071
MI_USER_INTERRUPT	1072
MI_USER_INTERRUPT	1073
MI_USER_INTERRUPT	1074
MI_USER_INTERRUPT	1075
MI_WAIT_FOR_EVENT	1076
Move	1081
Move Indexed	1083
Multiply	1085
Multiply Accumulate	1087
Multiply Accumulate High	1089
Multiply Add	1091
Multiply Add for Macro	1095
No Operation	1098
Oword Aligned Block Read MSD	1099
Oword Block Read MSD	1100
Oword Block Write MSD	1101
Oword Dual Block Read MSD	1102
Oword Dual Block Write MSD	1103
PIPE_CONTROL	1104
PIPELINE_SELECT	1112
Plane	1115
Read Surface Info MSD	1117
REP16 Render Target Write MSD	1118
Return	1121



Round Down	1123
Round to Nearest or Even	1124
Round to Zero	1126
Round Up	1127
Sampler Cache Media Block Read MSD	1128
Sampler Cache Oword Unaligned Block Read MSD	1129
Scaled Untyped Surface Read MSD	1130
Scaled Untyped Surface Write MSD	1131
Scattered Move	1132
Scratch Block Read MSD.....	1134
Scratch Block Write MSD.....	1136
Select	1137
Send Message.....	1139
SFC_AVS_CHROMA_Coeff_Table.....	1142
SFC_AVS_LUMA_Coeff_Table.....	1145
SFC_AVS_STATE	1150
SFC_FRAME_START	1152
SFC_IEF_STATE	1153
SFC_LOCK	1165
SFC_STATE.....	1167
Shift Left.....	1192
Shift Right.....	1194
SIMD8 Render Target Read MSD	1196
SIMD8 Render Target Write MSD	1199
SIMD16 Render Target Read MSD	1202
SIMD16 Render Target Write MSD	1205
Split Send Message.....	1208
STATE_BASE_ADDRESS	1213
STATE_PREFETCH	1221
STATE_SIP.....	1223
Sum of Absolute Difference 2.....	1224
Sum of Absolute Difference Accumulate 2.....	1226
Typed Surface Read MSD	1228



Typed Surface Write MSD	1229
Untyped Surface Read MSD	1230
Untyped Surface Write MSD	1231
URB Hword Dual Block Read MSD	1232
URB Hword Dual Block Write MSD	1233
URB Oword Block Write MSD	1234
URB Oword Dual Block Read MSD	1235
URB Oword Dual Block Write MSD	1236
VD_PIPELINE_FLUSH	1237
VEB_DI_IECP	1239
VEBOX_STATE	1248
VEBOX_SURFACE_STATE	1256
VEBOX_TILING_CONVERT	1264
Wait Notification	1266
While	1268
XY_COLOR_BLT	1270
XY_FAST_COPY_BLT	1272
XY_FULL_BLT	1275
XY_FULL_IMMEDIATE_PATTERN_BLT	1278
XY_FULL_MONO_PATTERN_BLT	1281
XY_FULL_MONO_PATTERN_MONO_SRC_BLT	1285
XY_FULL_MONO_SRC_BLT	1288
XY_FULL_MONO_SRC_IMMEDIATE_PATTERN_BLT	1291
XY_MONO_PAT_BLT	1294
XY_MONO_PAT_FIXED_BLT	1297
XY_MONO_SRC_COPY_BLT	1300
XY_MONO_SRC_COPY_IMMEDIATE_BLT	1303
XY_PAT_BLT	1306
XY_PAT_BLT_IMMEDIATE	1309
XY_PAT_CHROMA_BLT	1311
XY_PAT_CHROMA_BLT_IMMEDIATE	1313
XY_PIXEL_BLT	1316
XY_SCANLINES_BLT	1317



XY_SETUP_BLT	1318
XY_SETUP_CLIP_BLT	1321
XY_SETUP_MONO_PATTERN_SL_BLT	1322
XY_SRC_COPY_BLT	1325
XY_SRC_COPY_CHROMA_BLT	1328
XY_TEXT_BLT	1331
XY_TEXT_IMMEDIATE_BLT	1333



3DPRIMITIVE

3DPRIMITIVE		
Source:	RenderCS	
Length Bias:	2	
<p>The 3DPRIMITIVE command is used to submit 3D primitives to be processed by the 3D pipeline. Typically the processing results in rendering pixel data into the render targets, but this is not required.</p> <p>The parameters passed in this command are forwarded to the Vertex Fetch function. The Vertex Fetch function will use this information to generate vertex data structures and store them in the URB. These vertices are then passed down the 3D pipeline.</p>		
Programming Notes		
<p>If the threads spawned by this command are required to observe memory writes performed by threads spawned from a previous command, software must precede this command with a command that performs a (preferably pipelined) memory flush (e.g., 3D_PIPECONTROL).</p>		
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: 3h 3DPRIMITIVE Format: OpCode
	23:16	3D Command Sub Opcode
		Default Value: 0h 3DPRIMITIVE Format: OpCode
	15	Reserved
	14	Reserved
		Format: MBZ
	13	Reserved
		Format: MBZ
	12	Reserved
11	Reserved	
	Reserved	
	Reserved	
10	Indirect Parameter Enable	
	Format: Enable If set, the values in DW 2-5 are ignored and replaced by the current values of the corresponding	



3DPRIMITIVE					
	<p>3DPRIM_xxx MMIO registers:</p> <ul style="list-style-type: none"> • 3DPRIM_VERTEX_COUNT (instead of DW2: VertexCountPerInstance) • 3DPRIM_START_VERTEX (instead of DW3: StartVertexLocation) • 3DPRIM_INSTANCE_COUNT (instead of DW4: InstanceCount) • 3DPRIM_START_INSTANCE (instead of DW5: StartInstanceLocation) • 3DPRIM_BASE_VERTEX (instead of DW6: BaseVertexLocation) <p>Indirect Parameter Enable and End Offset Enable shall not be ENABLED at the same time, or behavior is UNDEFINED.</p>				
9	<p>UAV Coherency Required</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U1</td> </tr> </table> <p>SW will be required to set this bit if there is the possibility of sharing a UAV from a previous 3DPRIMITIVE command. If set, this command may cause a flush due to UAV coherency requirements. If none of the shaders have UAV access enabled, then this bit is ignored.</p>	Format:	U1		
Format:	U1				
8	<p>Predicate Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>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.</p>	Format:	Enable		
Format:	Enable				
7:0	<p>DWord Length</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Default Value:</td> <td>5h Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table>	Default Value:	5h Excludes DWord (0,1)	Format:	=n Total Length - 2
Default Value:	5h Excludes DWord (0,1)				
Format:	=n Total Length - 2				
1	<p>31:10 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>9 End Offset Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, the Vertex Count Per Instance field is IGNORED, and the 3DPRIM_END_OFFSET register is used to indirectly specify the vertex count by defining the amount of valid data in VB0. The following restrictions apply:</p> <ul style="list-style-type: none"> • VB0 must be enabled for use • VertexAccessType = SEQUENTIAL • Start Vertex Location = 0 • Start Instance Location = 0 • Base Vertex Location = 0 <p>Vertices are output until EndOffset is reached or exceeded in VB0. If EndOffset is reached or</p>	Format:	MBZ	Format:	Enable
Format:	MBZ				
Format:	Enable				



3DPRIMITIVE										
	<p>exceeded within the data associated with a vertex, that vertex is considered incomplete and will not be output. Partial objects will be discarded (as is normally done). If clear, End Offset is ignored. Indirect Parameter Enable and End Offset Enable must not be ENABLED at the same time, or behavior is UNDEFINED.</p>									
8	<p>Vertex Access Type This field specifies how data held in vertex buffers marked as VERTEXDATA is accessed by Vertex Fetch.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>SEQUENTIAL</td> <td>VERTEXDATA buffers are accessed sequentially. Require if End Offset Enable is ENABLED.</td> </tr> <tr> <td>1h</td> <td>RANDOM</td> <td>VERTEXDATA buffers are accessed randomly via an index obtained from the Index Buffer.</td> </tr> </tbody> </table>	Value	Name	Description	0h	SEQUENTIAL	VERTEXDATA buffers are accessed sequentially. Require if End Offset Enable is ENABLED.	1h	RANDOM	VERTEXDATA buffers are accessed randomly via an index obtained from the Index Buffer.
Value	Name	Description								
0h	SEQUENTIAL	VERTEXDATA buffers are accessed sequentially. Require if End Offset Enable is ENABLED.								
1h	RANDOM	VERTEXDATA buffers are accessed randomly via an index obtained from the Index Buffer.								
7:6	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
5:0	<p>Primitive Topology Type</p> <table border="1"> <tr> <td>Format:</td> <td>3D_Prim_Topo_Type See table below for encoding, see 3D Overview for diagrams and general comments</td> </tr> </table> <p style="text-align: center;">Description</p> <p>This field specifies the topology type of 3D primitive generated by this command. Note that a single primitive topology (list/strip/fan/etc.) can contain a number of basic objects (lines, triangles, etc.).</p> <p>This field is ignored. The topology type is specified via the 3DSTATE_VF_TOPOLOGY command.</p>	Format:	3D_Prim_Topo_Type See table below for encoding, see 3D Overview for diagrams and general comments							
Format:	3D_Prim_Topo_Type See table below for encoding, see 3D Overview for diagrams and general comments									
2	<p>31:0 Vertex Count Per Instance</p> <table border="1"> <tr> <td>Format:</td> <td>U32 Count of vertices</td> </tr> </table> <p>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, 2³²-1] (upper limit probably constrained by VB size) Ignored if End Offset Enable or Indirect Parameter Enable is ENABLED.</p> <p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> This per-instance value should specify a valid number of vertices for the primitive topology type. E.g., for 3DPRIM_TRILIST_ADJ, this field should specify a multiple of 6 vertices. However, in cases where too few or too many vertices are provided, the unused vertices will be silently discarded by the pipeline. A 0 value in this field effectively makes the command a 'no-operation'. 	Format:	U32 Count of vertices							
Format:	U32 Count of vertices									



		3DPRIMITIVE											
3	31:0	Start Vertex Location <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U32 structure index</td> </tr> </table> <p>This field specifies the "starting vertex" for each instance. This allows skipping over part of the vertices in a buffer if, for example, a previous 3DPRIMITIVE command had already drawn the primitives associated with the earlier entries.</p> <p>For SEQUENTIAL access, this field specifies, for each instance, a starting structure index into the vertex buffers</p> <p>For RANDOM access, this field specifies, for each instance, a starting index into the Index Buffer.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2"> <ul style="list-style-type: none"> 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 </td> </tr> </table>		Format:	U32 structure index	Programming Notes		<ul style="list-style-type: none"> 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 					
Format:	U32 structure index												
Programming Notes													
<ul style="list-style-type: none"> 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 													
4	31:0	Instance Count <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U32 Count of instances</td> </tr> </table> <p>This field specifies the number of instances by which the primitive topology is to be regenerated. A value of 0 indicates "no instances" (no-op operation). A value of 1 effectively specifies "non-instanced" operation, though vertex buffers will still be used to provide instance data, if so programmed.</p> <p>Ignored if Indirect Parameter Enable is ENABLED.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 60%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0, FFFFFFFFh]</td> <td></td> </tr> </tbody> </table>		Format:	U32 Count of instances	Value	Name	[0, FFFFFFFFh]					
Format:	U32 Count of instances												
Value	Name												
[0, FFFFFFFFh]													
5	31:0	Start Instance Location <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U32 structure index</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> <tr> <td colspan="2"> <p>This field specifies the "starting instance" for the command as an initial structure index into Vertex Buffers for vertex elements with InstancingEnable set.</p> <p>Subsequent instances will access sequential instance data structures, as controlled by the Instance Data Step Rate.</p> </td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2"> <ul style="list-style-type: none"> 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. </td> </tr> </table>		Format:	U32 structure index	Description		<p>This field specifies the "starting instance" for the command as an initial structure index into Vertex Buffers for vertex elements with InstancingEnable set.</p> <p>Subsequent instances will access sequential instance data structures, as controlled by the Instance Data Step Rate.</p>		Programming Notes		<ul style="list-style-type: none"> 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. 	
Format:	U32 structure index												
Description													
<p>This field specifies the "starting instance" for the command as an initial structure index into Vertex Buffers for vertex elements with InstancingEnable set.</p> <p>Subsequent instances will access sequential instance data structures, as controlled by the Instance Data Step Rate.</p>													
Programming Notes													
<ul style="list-style-type: none"> 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. 													



3DPRIMITIVE						
6	31:0	<p>Base Vertex Location</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>S31 index structure bias</td> </tr> </table> <p>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.</p> <table border="1" style="width: 100%;"> <tr> <th style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td> <ul style="list-style-type: none"> 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. </td> </tr> </table>	Format:	S31 index structure bias	Programming Notes	<ul style="list-style-type: none"> 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.
Format:	S31 index structure bias					
Programming Notes						
<ul style="list-style-type: none"> 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. 						



3DSTATE_AA_LINE_PARAMETERS

3DSTATE_AA_LINE_PARAMETERS			
Source:	RenderCS		
Length Bias:	2		
<p>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, versus in the previous iterations where U0.4 values were passed.</p>			
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:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	0Ah 3DSTATE_AA_LINE_PARAMETERS	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	Dword Length		
	Default Value:	1h Excludes Dword (0,1)	
	Format:	=n Total Length - 2	
1	31:24	AA Point Coverage Bias	
		Format:	U0.8
	This field specifies the bias term to be used in the aa coverage computation for edges 0 and 3.		
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	AA Point Coverage Slope		
	Format:	U0.8	
This field specifies the slope term to be used in the aa coverage computation for edges 0 and 3. If this field is zero, the Windower will revert to legacy aa line coverage computation (though still output expanded U0.8 coverage values).			



3DSTATE_AA_LINE_PARAMETERS				
2	7:0	<p>AA Coverage Slope</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.8</td> </tr> </table> <p>This field specifies the slope term to be used in the aa coverage computation for edges 0 and 3. If this field is zero, the Windower will revert to legacy aa line coverage computation (though still output expanded U0.8 coverage values).</p>	Format:	U0.8
	Format:	U0.8		
	31:24	<p>AA Point Coverage EndCap Bias</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.8</td> </tr> </table> <p>This field specifies the bias term to be used in the aa coverage computation for edges 1 and 2.</p>	Format:	U0.8
	Format:	U0.8		
	23:16	<p>AA Coverage EndCap Bias</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.8</td> </tr> </table> <p>This field specifies the bias term to be used in the aa coverage computation for edges 1 and 2.</p>	Format:	U0.8
Format:	U0.8			
15:8	<p>AA Point Coverage EndCap Slope</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.8</td> </tr> </table> <p>This field specifies the slope term to be used in the aa coverage computation for edges 1 and 2.</p>	Format:	U0.8	
Format:	U0.8			
7:0	<p>AA Coverage EndCap Slope</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.8</td> </tr> </table> <p>This field specifies the slope term to be used in the aa coverage computation for edges 1 and 2.</p>	Format:	U0.8	
Format:	U0.8			



3DSTATE_BINDING_TABLE_EDIT_DS

3DSTATE_BINDING_TABLE_EDIT_DS			
Source:	RenderCS		
Length Bias:	2		
This command edits the binding table for DS.			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		46h 3DSTATE_BINDING_TABLE_EDIT_DS	
Format:		OpCode	
15:9	Reserved		
	Format:	MBZ	
8:0	DWord Length		
	Format:	=n	
	Value	Name	
	0h	DWORD_COUNT_n [Default]	
	0h - 100h	Range	
1	31:16	Binding Table Block Clear	
		Format:	U16
	<p>Each bit in this field corresponds to a 16 entry block of the binding table. Bit 0 of this field corresponds to entries 0-15, bit 1 to 16-31, and so on. When a bit is set it clears the corresponding bind table entries to 0. (effectively disabling them). The clear is applied before the individual binding table entries contained in this message are applied. When this bit is clear then the corresponding 16 entry block is not cleared.</p>		
15:2	Reserved		
Format:		MBZ	



3DSTATE_BINDING_TABLE_EDIT_DS																	
	1:0	<p>Binding Table Edit Target Specifies which core should respond to this 3DSTATE_BINDING_TABLE_EDIT_DS command:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">11b</td> <td style="text-align: center;">All Cores</td> <td>All cores should respond to this command</td> </tr> <tr> <td style="text-align: center;">10b</td> <td style="text-align: center;">Core 1</td> <td>Only Core1 should respond to this command</td> </tr> <tr> <td style="text-align: center;">01b</td> <td style="text-align: center;">Core 0</td> <td>Only Core0 should respond to this command</td> </tr> <tr> <td style="text-align: center;">00b</td> <td style="text-align: center;">Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	11b	All Cores	All cores should respond to this command	10b	Core 1	Only Core1 should respond to this command	01b	Core 0	Only Core0 should respond to this command	00b	Reserved	Reserved
		Value	Name	Description													
		11b	All Cores	All cores should respond to this command													
		10b	Core 1	Only Core1 should respond to this command													
		01b	Core 0	Only Core0 should respond to this command													
00b	Reserved	Reserved															
2..n	31:0	<p>Entry [n]</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td style="text-align: center;">BINDING_TABLE_EDIT_ENTRY</td> </tr> </table>	Format:	BINDING_TABLE_EDIT_ENTRY													
Format:	BINDING_TABLE_EDIT_ENTRY																



3DSTATE_BINDING_TABLE_EDIT_GS

3DSTATE_BINDING_TABLE_EDIT_GS			
Source:	RenderCS		
Length Bias:	2		
This command edits the binding table for GS.			
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:	0h 3DSTATE_PIPELINED
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	44h 3DSTATE_BINDING_TABLE_EDIT_GS	
	Format:	OpCode	
15:9	Reserved		
	Format:	MBZ	
8:0	DWord Length		
	Format:	=n	
	Value	Name	
	0h	DWORD_COUNT_n [Default]	
0h - 100h	Range		
1	31:16	Binding Table Block Clear	
		Format:	U16
	<p>Each bit in this field corresponds to a 16 entry block of the binding table. Bit 0 of this field corresponds to entries 0-15, bit 1 to 16-31, and so on. When a bit is set it clears the corresponding bind table entries to 0. (effectively disabling them). The clear is applied before the individual binding table entries contained in this message are applied. When this bit is clear then the corresponding 16 entry block is not cleared.</p>		
15:2	Reserved		
Format:		MBZ	



3DSTATE_BINDING_TABLE_EDIT_GS																	
	1:0	Binding Table Edit Target Specifies which core should respond to this 3DSTATE_BINDING_TABLE_EDIT_GS command:															
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">11b</td> <td>All Cores</td> <td>All cores should respond to this command</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Core 1</td> <td>Only Core1 should respond to this command</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Core 0</td> <td>Only Core0 should respond to this command</td> </tr> <tr> <td style="text-align: center;">00b</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	11b	All Cores	All cores should respond to this command	10b	Core 1	Only Core1 should respond to this command	01b	Core 0	Only Core0 should respond to this command	00b	Reserved	Reserved
		Value	Name	Description													
		11b	All Cores	All cores should respond to this command													
		10b	Core 1	Only Core1 should respond to this command													
01b	Core 0	Only Core0 should respond to this command															
00b	Reserved	Reserved															
2..n	31:0	Entry [n] Format: BINDING_TABLE_EDIT_ENTRY															



3DSTATE_BINDING_TABLE_EDIT_HS

3DSTATE_BINDING_TABLE_EDIT_HS			
Source:	RenderCS		
Length Bias:	2		
This command edits the binding table for HS.			
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:	0h 3DSTATE_PIPELINED
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	45h 3DSTATE_BINDING_TABLE_EDIT_HS	
	Format:	OpCode	
15:9	Reserved		
	Format:	MBZ	
8:0	DWord Length		
	Format:	=n	
	Value	Name	
	0h	DWORD_COUNT_n [Default]	
	0h - 100h	Range	
1	31:16	Binding Table Block Clear	
		Format:	U16
	<p>Each bit in this field corresponds to a 16 entry block of the binding table. Bit 0 of this field corresponds to entries 0-15, bit 1 to 16-31, and so on. When a bit is set it clears the corresponding bind table entries to 0. (effectively disabling them). The clear is applied before the individual binding table entries contained in this message are applied. When this bit is clear then the corresponding 16 entry block is not cleared.</p>		
15:2	Reserved		
Format:		MBZ	



3DSTATE_BINDING_TABLE_EDIT_HS																	
	1:0	Binding Table Edit Target Specifies which core should respond to this 3DSTATE_BINDING_TABLE_EDIT_HS command:															
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>11b</td> <td>All Cores</td> <td>All cores should respond to this command</td> </tr> <tr> <td>10b</td> <td>Core 1</td> <td>Only Core1 should respond to this command</td> </tr> <tr> <td>01b</td> <td>Core 0</td> <td>Only Core0 should respond to this command</td> </tr> <tr> <td>00b</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	11b	All Cores	All cores should respond to this command	10b	Core 1	Only Core1 should respond to this command	01b	Core 0	Only Core0 should respond to this command	00b	Reserved	Reserved
		Value	Name	Description													
		11b	All Cores	All cores should respond to this command													
		10b	Core 1	Only Core1 should respond to this command													
01b	Core 0	Only Core0 should respond to this command															
00b	Reserved	Reserved															
2..n	31:0	Entry [n] Format: BINDING_TABLE_EDIT_ENTRY															



3DSTATE_BINDING_TABLE_EDIT_PS

3DSTATE_BINDING_TABLE_EDIT_PS			
Source:	RenderCS		
Length Bias:	2		
This command edits the binding table for PS.			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		47h 3DSTATE_BINDING_TABLE_EDIT_PS	
Format:		OpCode	
15:9	Reserved		
	Format:	MBZ	
8:0	DWord Length		
	Format:	=n	
	Value	Name	
	0h	DWORD_COUNT_n [Default]	
	0h - 100h	Range	
1	31:16	Binding Table Block Clear	
		Format:	U16
	<p>Each bit in this field corresponds to a 16 entry block of the binding table. Bit 0 of this field corresponds to entries 0-15, bit 1 to 16-31, and so on. When a bit is set it clears the corresponding bind table entries to 0. (effectively disabling them). The clear is applied before the individual binding table entries contained in this message are applied. When this bit is clear then the corresponding 16 entry block is not cleared.</p>		
15:2	Reserved		
Format:		MBZ	



3DSTATE_BINDING_TABLE_EDIT_PS																	
	1:0	Binding Table Edit Target Specifies which core should respond to this 3DSTATE_BINDING_TABLE_EDIT_PS command:															
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>11b</td> <td>All Cores</td> <td>All cores should respond to this command</td> </tr> <tr> <td>10b</td> <td>Core 1</td> <td>Only Core1 should respond to this command</td> </tr> <tr> <td>01b</td> <td>Core 0</td> <td>Only Core0 should respond to this command</td> </tr> <tr> <td>00b</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	11b	All Cores	All cores should respond to this command	10b	Core 1	Only Core1 should respond to this command	01b	Core 0	Only Core0 should respond to this command	00b	Reserved	Reserved
		Value	Name	Description													
		11b	All Cores	All cores should respond to this command													
		10b	Core 1	Only Core1 should respond to this command													
01b	Core 0	Only Core0 should respond to this command															
00b	Reserved	Reserved															
2..n	31:0	Entry [n] Format: BINDING_TABLE_EDIT_ENTRY															



3DSTATE_BINDING_TABLE_EDIT_VS

3DSTATE_BINDING_TABLE_EDIT_VS			
Source:	RenderCS		
Length Bias:	2		
This command edits the binding table for VS. The 3DSTATE_BINDING_TABLE_EDIT_VS is a variable length command.			
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: 0h 3DSTATE_PIPELINED Format: OpCode	
	23:16	3D Command Sub Opcode	
Default Value: 43h 3DSTATE_BINDING_TABLE_EDIT_VS Format: OpCode			
15:9	Reserved		
8:0	DWord Length	Format: =n	
		Value	Name
		0h	DWORD_COUNT_n [Default]
		0h - 100h	Range
1	31:16	Binding Table Block Clear	
		Format: U16 Each bit in this field corresponds to a 16 entry block of the binding table. Bit 0 of this field corresponds to entries 0-15, bit 1 to 16-31, and so on. When a bit is set it clears the corresponding bind table entries to 0. (affectively disabling them). The clear is applied before the individual binding table entries contained in this message are applied. When this bit is clear then the corresponding 16 entry block is not cleared.	
	15:2	Reserved	
		Format: MBZ	



3DSTATE_BINDING_TABLE_EDIT_VS																	
	1:0	<p>Binding Table Edit Target Specifies which core should respond to this 3DSTATE_BINDING_TABLE_EDIT_VS command:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">11b</td> <td style="text-align: center;">All Cores</td> <td>All cores should respond to this command</td> </tr> <tr> <td style="text-align: center;">10b</td> <td style="text-align: center;">Core 1</td> <td>Only Core1 should respond to this command</td> </tr> <tr> <td style="text-align: center;">01b</td> <td style="text-align: center;">Core 0</td> <td>Only Core0 should respond to this command</td> </tr> <tr> <td style="text-align: center;">00b</td> <td style="text-align: center;">Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	11b	All Cores	All cores should respond to this command	10b	Core 1	Only Core1 should respond to this command	01b	Core 0	Only Core0 should respond to this command	00b	Reserved	Reserved
		Value	Name	Description													
		11b	All Cores	All cores should respond to this command													
		10b	Core 1	Only Core1 should respond to this command													
		01b	Core 0	Only Core0 should respond to this command													
00b	Reserved	Reserved															
2..n	31:0	<p>Entry [n]</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>BINDING_TABLE_EDIT_ENTRY</td> </tr> </table>	Format:	BINDING_TABLE_EDIT_ENTRY													
Format:	BINDING_TABLE_EDIT_ENTRY																



3DSTATE_BINDING_TABLE_POINTERS_DS

3DSTATE_BINDING_TABLE_POINTERS_DS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_BINDING_TABLE_POINTERS_DS command is used to define the location of fixed functions' BINDING_TABLE_STATE. Only some of the fixed functions utilize binding tables.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	28h 3DSTATE_BINDING_TABLE_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:	SurfaceStateOffset[15:5]BINDING_TABLE_STATE*256 When Binding Table Pool is disabled		
Format:	SurfaceStateOffset[16:6]BINDING_TABLE_STATE*256 When Binding Table Pool is enabled		
<p>Specifies an aligned address offset of the function's BINDING_TABLE_STATE. The offset's base and alignment differ depending on whether HW Binding Table is enabled:</p> <p>If HW Binding Table is disabled, the offset is relative to Surface State Base Address and the alignment is 32B.</p> <p>If HW Binding Table is enabled the offset is relative to the Binding Table Pool Base Address and the alignment is 64B.</p>			



3DSTATE_BINDING_TABLE_POINTERS_DS				
	4:0	Reserved		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			



3DSTATE_BINDING_TABLE_POINTERS_GS

3DSTATE_BINDING_TABLE_POINTERS_GS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_BINDING_TABLE_POINTERS_GS command is used to define the location of fixed functions' BINDING_TABLE_STATE. Only some of the fixed functions utilize binding tables.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	29h 3DSTATE_BINDING_TABLE_POINTERS_GS	
	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 GS Binding Table	
Format:	SurfaceStateOffset[15:5]BINDING_TABLE_STATE*256 When Binding Table Pool is disabled		
Format:	SurfaceStateOffset[16:6]BINDING_TABLE_STATE*256 When Binding Table Pool is enabled		
<p>Specifies an aligned address offset of the function's BINDING_TABLE_STATE. The offset's base and alignment differ depending on whether HW Binding Table is enabled:</p> <p>If HW Binding Table is disabled, the offset is relative to Surface State Base Address and the alignment is 32B.</p> <p>If HW Binding Table is enabled the offset is relative to the Binding Table Pool Base Address and the alignment is 64B.</p>			



3DSTATE_BINDING_TABLE_POINTERS_GS				
	4:0	Reserved		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			



3DSTATE_BINDING_TABLE_POINTERS_HS

3DSTATE_BINDING_TABLE_POINTERS_HS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_BINDING_TABLE_POINTERS_HS command is used to define the location of fixed functions' BINDING_TABLE_STATE. Only some of the fixed functions utilize binding tables.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	27h 3DSTATE_BINDING_TABLE_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:	SurfaceStateOffset[15:5]BINDING_TABLE_STATE*256 When Binding Table Pool is disabled		
Format:	SurfaceStateOffset[16:6]BINDING_TABLE_STATE*256 When Binding Table Pool is enabled		
<p>Specifies an aligned address offset of the function's BINDING_TABLE_STATE. The offset's base and alignment differ depending on whether HW Binding Table is enabled:</p> <p>If HW Binding Table is disabled, the offset is relative to Surface State Base Address and the alignment is 32B.</p> <p>If HW Binding Table is enabled the offset is relative to the Binding Table Pool Base Address and the alignment is 64B.</p>			



3DSTATE_BINDING_TABLE_POINTERS_HS				
	4:0	Reserved		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			



3DSTATE_BINDING_TABLE_POINTERS_PS

3DSTATE_BINDING_TABLE_POINTERS_PS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_BINDING_TABLE_POINTERS_PS command is used to define the location of fixed functions' BINDING_TABLE_STATE. Only some of the fixed functions utilize binding tables.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	2Ah 3DSTATE_BINDING_TABLE_POINTERS_PS	
	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 PS Binding Table	
		Format:	SurfaceStateOffset[15:5]BINDING_TABLE_STATE*256 When Binding Table Pool is disabled
		Format:	SurfaceStateOffset[16:6]BINDING_TABLE_STATE*256 When Binding Table Pool is enabled
		Description	
			Specifies an aligned address offset of the function's BINDING_TABLE_STATE. The offset's base and alignment differ depending on whether HW Binding Table is enabled: If HW Binding Table is disabled, the offset is relative to Surface State Base Address and the alignment is 32B . If HW Binding Table is enabled the offset is relative to the Binding Table Pool Base Address and



3DSTATE_BINDING_TABLE_POINTERS_PS

		<p>the a398758)lignment is 64B.</p> <p>Specifies an aligned address offset of the function's BINDING_TABLE_STATE. The offset's base and alignment differ depending on whether HW Binding Table is enabled and the setting of HW Binding Table Alignment field: If HW Binding Table Pool is disabled and the HW Binding Table Alignment is not set to 256B, the offset is relative to Surface State Base Address and the alignment is 32B. If HW Binding Table Pool is enabled and the HW Binding Table Alignment is not set to 256B, the offset is relative to the Binding Table Pool Base Address and the alignment is 64B. If HW Binding Table Pool is disabled and the HW Binding Table Alignment is set to 256B, the offset is relative to the Surface State Base Address and the alignment is 256B. If HW Binding Table Pool is enabled and the HW Binding Table Alignment is set to 256B, the offset is relative to the Binding Table Pool Base Address and the alignment is 256B.</p>	
	4:0	Reserved	
		Format:	MBZ



3DSTATE_BINDING_TABLE_POINTERS_VS

3DSTATE_BINDING_TABLE_POINTERS_VS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_BINDING_TABLE_POINTERS_VS command is used to define the location of fixed functions' BINDING_TABLE_STATE. Only some of the fixed functions utilize binding tables.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	26h 3DSTATE_BINDING_TABLE_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:	SurfaceStateOffset[15:5]BINDING_TABLE_STATE*256 When Binding Table Pool is disabled		
Format:	SurfaceStateOffset[16:6]BINDING_TABLE_STATE*256 When Binding Table Pool is enabled		
<p>Specifies an aligned address offset of the function's BINDING_TABLE_STATE. The offset's base and alignment differ depending on whether HW Binding Table is enabled:</p> <p>If HW Binding Table is disabled, the offset is relative to Surface State Base Address and the alignment is 32B.</p> <p>If HW Binding Table is enabled the offset is relative to the Binding Table Pool Base Address and the alignment is 64B.</p>			



3DSTATE_BINDING_TABLE_POINTERS_VS		
	4:0	Reserved
	Format:	MBZ



3DSTATE_BINDING_TABLE_POOL_ALLOC

3DSTATE_BINDING_TABLE_POOL_ALLOC			
Source:	RenderCS		
Length Bias:	2		
Description			
This command sets up the binding table pool for HW generated binding tables.			
Programming Notes			
When RS is enabled due to a MI_RS_CONTROL or MI_BATCH_BUFFER_START with RS enable bit set, driver must reprogram the 3DSTATE_BINDING_TABLE_POOL_ALLOC to ensure the resource streamer and render engine are in sync with the programming with the command. Otherwise there could be cases where RS sees that the Binding Table Pool is disabled while the render pipeline sees the binding table is enabled in the case the 3DSTATE_BINDING_TABLE_POOL_ALLOC was enabled while RS was off.			
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:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	19h 3DSTATE_BINDING_TABLE_POOL_ALLOC	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Format:	=n	
	Value	Name	
	2h	DWORD_COUNT_n [Default]	
1..2	63:12	Binding Table Pool Base Address	
		Format:	GraphicsAddress63-12
		This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space.	



3DSTATE_BINDING_TABLE_POOL_ALLOC											
	11	Binding Table Pool Enable Format: U1 When this bit is set it enables HW generation of binding tables. When this bit is cleared it disables HW generation of binding tables.									
	10	Reserved									
	9:7	Reserved									
	6:0	Surface Object Control State Format: MEMORY_OBJECT_CONTROL_STATE Specifies the memory object control state for this surface. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> Bit 2 is not programmable and is always zero.									
3	31:12	Binding Table Pool Buffer Size Format: U20 This field specifies the size of the buffer in 4K pages. Any access which straddle or go past the end of the buffer will return 0. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,1048575]</td> <td></td> <td></td> </tr> <tr> <td style="text-align: center;">0</td> <td>No Valid Data</td> <td>There is no valid data in the buffer</td> </tr> </tbody> </table> <div style="text-align: center; background-color: #e6f2ff; padding: 2px; margin-top: 5px;">Restriction</div> Restriction : Programming size of zero is illegal in the case that the pool is enabled.	Value	Name	Description	[0,1048575]			0	No Valid Data	There is no valid data in the buffer
	Value	Name	Description								
[0,1048575]											
0	No Valid Data	There is no valid data in the buffer									
11:0	Reserved Format: MBZ										



3DSTATE_BLEND_STATE_POINTERS

3DSTATE_BLEND_STATE_POINTERS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_BLEND_STATE_POINTERS command is used to set up the pointers to the color calculator state.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	24h 3DSTATE_BLEND_STATE_POINTERS	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h DWORD_COUNT_n	
	Format:	=n	
1	31:6	Blend State Pointer	
		Format: DynamicStateOffset[31:6]BLEND_STATE*8	
	Specifies the 64-byte aligned offset of the BLEND_STATE. This offset is relative to the Dynamic State Base Address .		
5:1	Reserved		
	Format:	MBZ	
0	Blend State Pointer Valid		
	Format: Enable		
This bit, if set, indicates that the BLEND_STATE pointer has changed and new state needs to be fetched.			



3DSTATE_CC_STATE_POINTERS

3DSTATE_CC_STATE_POINTERS		
Source:	RenderCS	
Length Bias:	2	
The 3DSTATE_CC_STATE_POINTERS command is used to set up the pointers to the color calculator state.		
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: 0h 3DSTATE_PIPELINED		
Format: OpCode		
23:16	3D Command Sub Opcode	
	Default Value: 0Eh 3DSTATE_CC_STATE_POINTERS	
	Format: OpCode	
15:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Default Value: 0h DWORD_COUNT_n	
	Format: =n	
1	31:6	Color Calc State Pointer
		Format: DynamicStateOffset[31:6]COLOR_CALC_STATE
	Specifies the 64-byte aligned offset of the COLOR_CALC_STATE. This offset is relative to the Dynamic State Base Address .	
5:1	Reserved	
	Format: MBZ	
0	Color Calc State Pointer Valid	
	Format: Enable	
If set, the hardware will fetch the CC state. This bit is context saved and restored so the CC state is considered undefined once this bit is cleared due to the possibility of the CC state changing between context switches.		



3DSTATE_CHROMA_KEY

3DSTATE_CHROMA_KEY			
Source:	RenderCS		
Length Bias:	2		
<p>The 3DSTATE_CHROMA_KEY instruction is used to program texture color/chroma-key key values. A table containing four set of values is supported. The ChromaKey Index sampler state variable is used to select which table entry is associated with the map. Texture chromakey functions are enabled and controlled via use of the ChromaKey Enable texture sampler state variable. Texture Color Key (keying on a paletted texture index) is not supported.</p>			
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:		Opcode	
23:16	3D Command Sub Opcode		
	Default Value:	04h 3DSTATE_CHROMA_KEY	
	Format:	Opcode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	2h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31:30	ChromaKey Table Index	
		Format:	U2 index
		Selects which entry in the ChromaKey table is to be loaded	
		Value	Name
		[0,3]	
29:0	Reserved		
	Format:	MBZ	
2	31:0	ChromaKey Low Value This field specifies the "low" (minimum) value of the chroma key range. Texel samples are	



3DSTATE_CHROMA_KEY																
	considered "matching the key" if each component of the texel falls within the (inclusive) chroma range. See ChromaKey High Value for further format, programming info.															
3	<p>31:0 ChromaKey High Value This field specifies the "high" (maximum) value of the chroma key range. Texel samples are considered "matching the key" if each component of the texel falls within the (inclusive) chroma range.</p> <p style="text-align: center;">Programming Notes</p> <p>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).</p> <p>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=0h 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.</p> <p>For channels in SNORM format in the surface format, the value in the high/low value for that channel 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. R5G5_SNORM_B6_UNORM), the ChromaKey is programmed as if all channels are SNORM.</p> <p>YUV ChromaKey will use an interpolated chrominance value from the map for comparison to the chroma key values for those texels without chrominance due to downsampling. The chrominance value used is the average of values to the left and right of the texel in question.</p> <p>It is UNDEFINED to program any component of the ChromaKey High Value to be less than the corresponding component of ChromaKey Low Value.</p> <p>Format = interpreted according to associated texel format "class":</p> <p>Only the surface formats listed as supported for chroma key in the surface formats table can be used with this feature. Use of any other surface format with chroma key enabled is UNDEFINED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Surface Format</th> <th style="text-align: center;">31:24</th> <th style="text-align: center;">23:15</th> <th style="text-align: center;">16:8</th> <th style="text-align: center;">7:0</th> </tr> </thead> <tbody> <tr> <td>ARGB and BC (DXT) formats</td> <td style="text-align: center;">A</td> <td style="text-align: center;">R</td> <td style="text-align: center;">G</td> <td style="text-align: center;">B</td> </tr> <tr> <td>YCrCb formats</td> <td style="text-align: center;">A</td> <td style="text-align: center;">Cr</td> <td style="text-align: center;">Y</td> <td style="text-align: center;">Cb</td> </tr> </tbody> </table>	Surface Format	31:24	23:15	16:8	7:0	ARGB and BC (DXT) formats	A	R	G	B	YCrCb formats	A	Cr	Y	Cb
Surface Format	31:24	23:15	16:8	7:0												
ARGB and BC (DXT) formats	A	R	G	B												
YCrCb formats	A	Cr	Y	Cb												



3DSTATE_CLEAR_PARAMS

3DSTATE_CLEAR_PARAMS			
Source:	RenderCS		
Length Bias:	2		
Description			
This command defines the depth clear value delivered as a pipelined state command. However, the state change pipelining isn't completely transparent (see restriction below).			
HW will internally manage the draining pipe and flushing of the caches when this command is issued. The PIPE_CONTROL restrictions are removed.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	04h 3DSTATE_CLEAR_PARAMS	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	Dword Length		
	Default Value:	1h Excludes Dword (0,1)	
	Format:	=n Total Length - 2	
1	31:0	Depth Clear Value	
		Format:	IEEE_Float
	This field defines the clear value that will be applied to the depth buffer if the Depth Buffer Clear field is enabled. It is valid only if Depth Buffer Clear Value Valid is set.		
	Programming Notes The clear value must be between the min and max depth values (inclusive) defined in the CC_VIEWPORT. If the depth buffer format is D32_FLOAT, then values must be limited to the range of +0.0f and 1.0f inclusive; values outside this range are reserved.		
2	31:1	Reserved	



3DSTATE_CLEAR_PARAMS			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
0	<p>Depth Clear Value Valid</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Boolean</td> </tr> </table> <p>This field enables the Depth Clear Value. If clear, the depth clear value is obtained from interpolated depth of an arbitrary pixel of the primitive rendered with Depth Buffer Clear set in WM_STATE or 3DSTATE_WM. If set, the depth clear value is obtained from the Depth Clear Value field of this command.</p>	Format:	Boolean
Format:	Boolean		



3DSTATE_CLIP

3DSTATE_CLIP			
Source:		RenderCS	
Length 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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		12h 3DSTATE_CLIP	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	02h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31:21	Reserved	
		Format:	MBZ
	20	Force User Clip Distance Cull Test Enable Bitmask	
Format:		Enable	
This field provides a work around override for the computation of SOL_INT::Render_Enable			
Value	Name	Description	
0h	Normal	Clip_INT::User Clip Distance Cull Test Enable Bitmask normally	
1h	Force	Forces Clip_INT::User Clip Distance Cull Test Enable Bitmask to use the value in 3DSTATE_CLIP:: User Clip Distance Cull Test Enable Bitmask	



3DSTATE_CLIP		
19	Vertex Sub Pixel Precision Select	
	Format:	U1
	Selects the number of fractional bits maintained in the vertex data	
	Value	Name
0h	8 Bit	8 sub pixel precision bits maintained
1h	4 Bit	4 sub pixel precision bits maintained
18	Early Cull Enable	
	Format:	Enable
	This field is used to enable/disable the EarlyCull function. When this bit is set triangles are checked if they are backface culled before proceedign through must clip function.	
	Programming Notes	
Setting this bit must not impact functionality, this state only controls the performance of the must clip function.		
Vertex Sub Pixel Precision Selectprecision must be set to "8 bit" in order avoid precision issues.		
17	Force User Clip Distance Clip Test Enable Bitmask	
	Format:	Enable
	This field provides a work around override for the computation of SOL_INT::Render_Enable.	
	Value	Name
0b	Normal	Clip_INT:: User Clip Distance Clip Test Enable Bitmask normally
1b	Force	Forces Clip_INT:: User Clip Distance Clip Test Enable Bitmask to use the value in 3DSTATE_CLIP::User Clip Distance Clip Test Enable Bitmask
16	Force Clip Mode	
	Format:	Enable
	This field provides a work around override for the computation of SOL_INT::Render_Enable.	
	Value	Name
0b	Normal	Clip_INT::Clip Mode is computed normally.
1b	Force	Forces Clip_INT::Clip Mode to use the value in 3DSTATE_CLIP::User Clip Mode.
15:11	Reserved	
	Format:	MBZ
10	Clipper Statistics Enable	
	Format:	Enable
	This bit controls whether Clip-unit-specific statistics register(s) can be incremented.	
	Value	Name
0h	Disable	CL_INVOCATIONS_COUNT cannot increment
1h	Enable	CL_INVOCATIONS_COUNT can increment
9:8	Reserved	
	Format:	MBZ



3DSTATE_CLIP								
	7:0	<p>User Clip Distance Cull Test Enable Bitmask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable[8]</td> </tr> </table> <p>This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept determination needs to be made (does not cause a must clip).DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.</p>	Format:	Enable[8]				
Format:	Enable[8]							
2	31	<p>Clip Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p>Specifies whether the Clip function is enabled or disabled (pass-through).</p>	Format:	Enable				
	Format:	Enable						
	30	<p>API Mode Controls the definition of the NEAR clipping plane</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>OGL</td> <td>NEAR VP boundary == 0.0 (NDC)</td> </tr> </tbody> </table>	Value	Name	Description	0h	OGL	NEAR VP boundary == 0.0 (NDC)
	Value	Name	Description					
	0h	OGL	NEAR VP boundary == 0.0 (NDC)					
	29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ						
	28	<p>Viewport XY Clip Test Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field is used to control whether the Viewport X, Y extents [-1,1] are considered in VertexClipTest. If both the Guardband and Viewport XY ClipTest are DISABLED, all vertices are considered "visible" with respect to the XY directions.</p>	Format:	Enable				
Format:	Enable							
27	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ							
26	<p>Guardband Clip Test Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field is used to control whether the Guardband X, Y extents are considered in VertexClipTest for non-point objects. If the Guardband ClipTest is DISABLED but the Viewport XY ClipTest is ENABLED, ClipDetermination operates as if the Guardband were coincident with the Viewport. If both the Guardband and Viewport XY ClipTest are DISABLED, all vertices are considered "visible" with respect to the XY directions.</p>	Format:	Enable					
Format:	Enable							
25:24	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ							
23:16	<p>User Clip Distance Clip Test Enable Bitmask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable[8]</td> </tr> </table> <p>This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept / must clip determination needs to be made.DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.</p>	Format:	Enable[8]					
Format:	Enable[8]							



3DSTATE_CLIP

15:13	Clip Mode	<p>This field specifies a general mode of the CLIP unit, when the CLIP unit is ENABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NORMAL</td> <td>TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>2h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>3h</td> <td>REJECT_ALL</td> <td>All objects are discarded</td> </tr> <tr> <td>4h</td> <td>ACCEPT_ALL</td> <td>All objects (except BAD objects) are trivially accepted. This effectively disables the clip-test/clip-determination function. Note that the CLIP unit will still filter out adjacency information, which may be required since the SF unit does not accept primitives with adjacency.</td> </tr> <tr> <td>5h-7h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>		Value	Name	Description	0h	NORMAL	TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded	1h	Reserved		2h	Reserved		3h	REJECT_ALL	All objects are discarded	4h	ACCEPT_ALL	All objects (except BAD objects) are trivially accepted. This effectively disables the clip-test/clip-determination function. Note that the CLIP unit will still filter out adjacency information, which may be required since the SF unit does not accept primitives with adjacency.	5h-7h	Reserved	
Value	Name	Description																						
0h	NORMAL	TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded																						
1h	Reserved																							
2h	Reserved																							
3h	REJECT_ALL	All objects are discarded																						
4h	ACCEPT_ALL	All objects (except BAD objects) are trivially accepted. This effectively disables the clip-test/clip-determination function. Note that the CLIP unit will still filter out adjacency information, which may be required since the SF unit does not accept primitives with adjacency.																						
5h-7h	Reserved																							
12:10	Reserved	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ																			
Format:	MBZ																							
9	Perspective Divide Disable	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Disable</td> </tr> </table> <p>This field disables the Perspective Divide function performed on homogeneous position read 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, the 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 SF unit (e.g., by clipping in CPU SW before submitting the objects).</p>		Format:	Disable																			
Format:	Disable																							
8	Non-Perspective Barycentric Enable	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables computation of non-perspective barycentric parameters in the clipper, which are sent to SF unit in the must clip case. This field must be enabled if any non-perspective interpolation modes are used in pixel shader.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>This field must be set whenever Enable bits 3 or 4 or 5 of 3DSTATE_WM:Barycentric Interpolation Mode is set. This indicates that one of the Non-perspective barycentric interpolation modes are used.</td> </tr> </tbody> </table>		Format:	Enable	Programming Notes	This field must be set whenever Enable bits 3 or 4 or 5 of 3DSTATE_WM:Barycentric Interpolation Mode is set. This indicates that one of the Non-perspective barycentric interpolation modes are used.																	
Format:	Enable																							
Programming Notes																								
This field must be set whenever Enable bits 3 or 4 or 5 of 3DSTATE_WM:Barycentric Interpolation Mode is set. This indicates that one of the Non-perspective barycentric interpolation modes are used.																								
7:6	Reserved	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ																			
Format:	MBZ																							



3DSTATE_CLIP											
5:4	Triangle Strip/List Provoking Vertex Select Format: U2 <hr/> enumerated type This field selects which vertex of a triangle (in a triangle strip or list primitive) is considered the "provoking vertex". <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0</td> </tr> <tr> <td>1h</td> <td>1</td> </tr> <tr> <td>2h</td> <td>2</td> </tr> <tr> <td>3h</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	0h	0	1h	1	2h	2	3h	Reserved
	Value	Name									
	0h	0									
1h	1										
2h	2										
3h	Reserved										
3:2	Line Strip/List Provoking Vertex Select Format: U2 <hr/> enumerated type This field selects which vertex of a line (in a line strip or list primitive) is considered the "provoking vertex". <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0</td> </tr> <tr> <td>1h</td> <td>1</td> </tr> <tr> <td>2h</td> <td>Reserved</td> </tr> <tr> <td>3h</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	0h	0	1h	1	2h	Reserved	3h	Reserved
	Value	Name									
	0h	0									
1h	1										
2h	Reserved										
3h	Reserved										
1:0	Triangle Fan Provoking Vertex Select Format: U2 <hr/> enumerated type This field selects which vertex of a triangle (in a triangle fan primitive) is considered the "provoking vertex". <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0</td> </tr> <tr> <td>1h</td> <td>1</td> </tr> <tr> <td>2h</td> <td>2</td> </tr> <tr> <td>3h</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	0h	0	1h	1	2h	2	3h	Reserved
	Value	Name									
	0h	0									
1h	1										
2h	2										
3h	Reserved										
3	31:28 Reserved Format: MBZ										



3DSTATE_CLIP			
27:17	<p>Minimum Point Width</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>U8.3 pixels</td> </tr> </table> <p>This value is used to clamp read-back PointWidth values.</p>	Format:	U8.3 pixels
Format:	U8.3 pixels		
16:6	<p>Maximum Point Width</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>U8.3 pixels</td> </tr> </table> <p>This value is used to clamp read-back PointWidth values.</p>	Format:	U8.3 pixels
Format:	U8.3 pixels		
5	<p>Force Zero RTA Index Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, the Clip unit will ignore the read-back RTAIndex and operate as if the value 0 was read-back. If clear, the read-back value is used.</p>	Format:	Enable
Format:	Enable		
4	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
3:0	<p>Maximum VP Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>U4-1 index value (# of viewports)</td> </tr> </table> <p>This field specifies the maximum valid VPIndex value, corresponding to the number of active viewports. If the source of the VPIndex exceeds this maximum value, a VPIndex value of 0 is passed down the pipeline. Note that this clamping does not affect a VPIndex value stored in the URB.</p>	Format:	U4-1 index value (# of viewports)
Format:	U4-1 index value (# of viewports)		



3DSTATE_CONSTANT_DS

3DSTATE_CONSTANT_DS			
Source:	RenderCS		
Length Bias:	2		
<p>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).</p>			
Programming Notes			
<p>3DSTATE_GATHER_DS command must be dispatched along with any 3DSTATE_CONSTANT_DS command when Gather Pool is enabled within a batch buffer.</p> <p>The 3DSTATE_CONSTANT_* command is not committed to the shader unit until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command is parsed. For example, the 3DSTATE_CONSTANT_VS command will not fetch the constant buffers from memory and make available to the shader until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the render command streamer. In case of multiple 3DSTATE_CONSTANT_VS programmed prior to 3DSTATE_BINDING_TABLE_POINTER_VS, only the most recently programmed 3DSTATE_CONSTANT_VS will be committed.</p> <p>On usage model of enabling legacy mode is when Resource Streamer is not enabled.</p>			
Workaround			
<p>Workaround:</p> <p>The driver must ensure The following case does not occur without a flush to the 3D engine: 3DSTATE_CONSTANT_* with buffer 3 read length equal to zero committed followed by a 3DSTATE_CONSTANT_* with buffer 0 read length not equal to zero committed. Possible ways to avoid this condition include:</p> <ul style="list-style-type: none"> • always force buffer 3 to have a non zero read length • always force buffer 0 to a zero read length 			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode



3DSTATE_CONSTANT_DS						
	23:16	3D Command Sub Opcode				
		Default Value: 1Ah 3DSTATE_CONSTANT_DS				
		Format: OpCode				
	15	Reserved				
		Format: MBZ				
	14:8	Constant Buffer Object Control State				
		Format: MEMORY_OBJECT_CONTROL_STATE				
		Specifies the memory object control state for all constant buffers defined in this command.				
	7:0	DWord Length				
		Format: =n Total Length - 2				
		<table border="1" style="width: 100%;"><thead><tr><th style="text-align: center;">Value</th><th style="text-align: center;">Name</th></tr></thead><tbody><tr><td style="text-align: center;">9h</td><td>Excludes DWord (0,1) [Default]</td></tr></tbody></table>	Value	Name	9h	Excludes DWord (0,1) [Default]
Value	Name					
9h	Excludes DWord (0,1) [Default]					
1..10	319:0	Constant Body				
		Format: 3DSTATE_CONSTANT(Body)				
		See the 3DSTATE_CONSTANT(Body) format for the shared portion of the 3DSTATE_CONSTANT command for VS, HS, DS, and GS.				



3DSTATE_CONSTANT_GS

3DSTATE_CONSTANT_GS			
Source:	RenderCS		
Length Bias:	2		
<p>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).</p>			
Programming Notes			
<p>A 3DSTATE_GATHER_GS command must be dispatched along with any 3DSTATE_CONSTANT_GS command when the Gather Pool is enabled within a batch buffer.</p>			
<p>The 3DSTATE_CONSTANT_* command is not committed to the shader unit until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command is parsed. For example, the 3DSTATE_CONSTANT_VS command will not fetch the constant buffers from memory and make available to the shader until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the render command streamer. In case of multiple 3DSTATE_CONSTANT_VS programmed prior to 3DSTATE_BINDING_TABLE_POINTER_VS, only the most recently programmed 3DSTATE_CONSTANT_VS will be committed.</p> <p>On usage model of enabling legacy mode is when Resource Streamer is not enabled.</p>			
Workaround			
<p>Workaround:</p> <p>The driver must ensure The following case does not occur without a flush to the 3D engine: 3DSTATE_CONSTANT_* with buffer 3 read length equal to zero committed followed by a 3DSTATE_CONSTANT_* with buffer 0 read length not equal to zero committed. Possible ways to avoid this condition include:</p> <ul style="list-style-type: none"> • always force buffer 3 to have a non zero read length • always force buffer 0 to a zero read length 			
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:	0h 3DSTATE_PIPELINED	
	Format:	OpCode	



3DSTATE_CONSTANT_GS							
	23:16	3D Command Sub Opcode <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>16h 3DSTATE_CONSTANT_GS</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	16h 3DSTATE_CONSTANT_GS	Format:	OpCode	
	Default Value:	16h 3DSTATE_CONSTANT_GS					
	Format:	OpCode					
	15	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ						
14:8	Constant Buffer Object Control State <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for all constant buffers defined in this command.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE				
Format:	MEMORY_OBJECT_CONTROL_STATE						
7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>9h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>	Format:	=n Total Length - 2	Value	Name	9h	Excludes DWord (0,1) [Default]
Format:	=n Total Length - 2						
Value	Name						
9h	Excludes DWord (0,1) [Default]						
1..10	319:0	Constant Body <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>3DSTATE_CONSTANT(Body)</td> </tr> </table> <p>Following table is the shared portion of the 3DSTATE_CONSTANT command for VS, HS, DS, and GS</p>	Format:	3DSTATE_CONSTANT(Body)			
Format:	3DSTATE_CONSTANT(Body)						



3DSTATE_CONSTANT_HS

3DSTATE_CONSTANT_HS			
Source:	RenderCS		
Length Bias:	2		
<p>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).</p>			
Programming Notes			
<p>A 3DSTATE_GATHER_HS command must be dispatched along with any 3DSTATE_CONSTANT_HS command when Gather Pool is enabled within a batch buffer.</p>			
<p>The 3DSTATE_CONSTANT_* command is not committed to the shader unit until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command is parsed. For example, the 3DSTATE_CONSTANT_VS command will not fetch the constant buffers from memory and make available to the shader until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the render command streamer. In case of multiple 3DSTATE_CONSTANT_VS programmed prior to 3DSTATE_BINDING_TABLE_POINTER_VS, only the most recently programmed 3DSTATE_CONSTANT_VS will be committed.</p> <p>On usage model of enabling legacy mode is when Resource Streamer is not enabled.</p>			
Workaround			
<p>Workaround:</p> <p>The driver must ensure The following case does not occur without a flush to the 3D engine: 3DSTATE_CONSTANT_* with buffer 3 read length equal to zero committed followed by a 3DSTATE_CONSTANT_* with buffer 0 read length not equal to zero committed. Possible ways to avoid this condition include:</p> <ul style="list-style-type: none"> • always force buffer 3 to have a non zero read length • always force buffer 0 to a zero read length 			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode



3DSTATE_CONSTANT_HS							
	23:16	3D Command Sub Opcode <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>19h 3DSTATE_CONSTANT_HS</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	19h 3DSTATE_CONSTANT_HS	Format:	OpCode	
	Default Value:	19h 3DSTATE_CONSTANT_HS					
	Format:	OpCode					
	15	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ						
14:8	Constant Buffer Object Control State <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for all constant buffers defined in this command.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE				
Format:	MEMORY_OBJECT_CONTROL_STATE						
7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">9h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>	Format:	=n Total Length - 2	Value	Name	9h	Excludes DWord (0,1) [Default]
Format:	=n Total Length - 2						
Value	Name						
9h	Excludes DWord (0,1) [Default]						
1..10	319:0	Constant Body <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>3DSTATE_CONSTANT(Body)</td> </tr> </table> <p>Following table is the shared portion of the 3DSTATE_CONSTANT command for VS, HS, DS, and GS</p>	Format:	3DSTATE_CONSTANT(Body)			
Format:	3DSTATE_CONSTANT(Body)						



3DSTATE_CONSTANT_PS

3DSTATE_CONSTANT_PS			
Source:	RenderCS		
Length Bias:	2		
<p>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).</p>			
Programming Notes			
<p>A 3DSTATE_GATHER_PS command must be dispatched along with any 3DSTATE_CONSTANT_PS command when the Gather Pool is enabled within a batch buffer.</p>			
<p>The 3DSTATE_CONSTANT_* command is not committed to the shader unit until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command is parsed. For example, the 3DSTATE_CONSTANT_VS command will not fetch the constant buffers from memory and make available to the shader until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the render command streamer. In case of multiple 3DSTATE_CONSTANT_VS programmed prior to 3DSTATE_BINDING_TABLE_POINTER_VS, only the most recently programmed 3DSTATE_CONSTANT_VS will be committed.</p> <p>On usage model of enabling legacy mode is when Resource Streamer is not enabled.</p>			
Workaround			
<p>Workaround:</p> <p>The driver must ensure The following case does not occur without a flush to the 3D engine: 3DSTATE_CONSTANT_* with buffer 3 read length equal to zero committed followed by a 3DSTATE_CONSTANT_* with buffer 0 read length not equal to zero committed. Possible ways to avoid this condition include:</p> <ul style="list-style-type: none"> • always force buffer 3 to have a non zero read length • always force buffer 0 to a zero read length 			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode



3DSTATE_CONSTANT_PS							
	23:16	3D Command Sub Opcode <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>17h 3DSTATE_CONSTANT_PS</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	17h 3DSTATE_CONSTANT_PS	Format:	OpCode	
	Default Value:	17h 3DSTATE_CONSTANT_PS					
	Format:	OpCode					
	15	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ						
14:8	Constant Buffer Object Control State <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for all constant buffers defined in this command.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE				
Format:	MEMORY_OBJECT_CONTROL_STATE						
7:0	Dword Length <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>9h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>	Format:	=n Total Length - 2	Value	Name	9h	Excludes DWord (0,1) [Default]
Format:	=n Total Length - 2						
Value	Name						
9h	Excludes DWord (0,1) [Default]						
1..10	319:0	Constant Body <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>3DSTATE_CONSTANT(Body)</td> </tr> </table> <p>Following table is the shared portion of the 3DSTATE_CONSTANT command for VS, HS, DS, and GS</p>	Format:	3DSTATE_CONSTANT(Body)			
Format:	3DSTATE_CONSTANT(Body)						



3DSTATE_CONSTANT_VS

3DSTATE_CONSTANT_VS			
Source:	RenderCS		
Length Bias:	2		
<p>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).</p>			
Programming Notes			
<p>A 3DSTATE_GATHER_VS command must be dispatched along with any 3DSTATE_CONSTANT_VS command when Gather Pool is enabled within a batch buffer.</p>			
<p>The 3DSTATE_CONSTANT_* command is not committed to the shader unit until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command is parsed. For example, the 3DSTATE_CONSTANT_VS command will not fetch the constant buffers from memory and make available to the shader until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the render command streamer. In case of multiple 3DSTATE_CONSTANT_VS programmed prior to 3DSTATE_BINDING_TABLE_POINTER_VS, only the most recently programmed 3DSTATE_CONSTANT_VS will be committed.</p> <p>On usage model of enabling legacy mode is when Resource Streamer is not enabled.</p>			
Workaround			
<p>Workaround:</p> <p>The driver must ensure The following case does not occur without a flush to the 3D engine: 3DSTATE_CONSTANT_* with buffer 3 read length equal to zero committed followed by a 3DSTATE_CONSTANT_* with buffer 0 read length not equal to zero committed. Possible ways to avoid this condition include:</p> <ul style="list-style-type: none"> • always force buffer 3 to have a non zero read length • always force buffer 0 to a zero read length 			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode



3DSTATE_CONSTANT_VS							
	23:16	3D Command Sub Opcode <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>15h 3DSTATE_CONSTANT_VS</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	15h 3DSTATE_CONSTANT_VS	Format:	OpCode	
	Default Value:	15h 3DSTATE_CONSTANT_VS					
	Format:	OpCode					
	15	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ						
14:8	Constant Buffer Object Control State <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for all constant buffers defined in this command.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE				
Format:	MEMORY_OBJECT_CONTROL_STATE						
7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">9h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>	Format:	=n Total Length - 2	Value	Name	9h	Excludes DWord (0,1) [Default]
Format:	=n Total Length - 2						
Value	Name						
9h	Excludes DWord (0,1) [Default]						
1..10	319:0 Constant Body <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>3DSTATE_CONSTANT(Body)</td> </tr> </table> <p>Following table is the shared portion of the 3DSTATE_CONSTANT command for VS, HS, DS, and GS</p>	Format:	3DSTATE_CONSTANT(Body)				
Format:	3DSTATE_CONSTANT(Body)						



3DSTATE_DEPTH_BUFFER

3DSTATE_DEPTH_BUFFER				
Source:	RenderCS			
Length Bias:	2			
<p>The depth buffer surface state is delivered as a pipelined state packet. However, the state change pipelining isn't completely transparent (see restriction below).</p> <p>WM HW will internally manage the draining pipe and flushing of the caches when this commands is issued. The PIPE_CONTROL restrictions are removed.</p>				
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:		0h 3DSTATE_PIPELINED		
Format:		OpCode		
23:16	3D Command Sub Opcode			
	Default Value:	5h 3DSTATE_DEPTH_BUFFER		
	Format:	OpCode		
15:8	Reserved			
	Format:	MBZ		
7:0	DWord Length			
	Format:	=n		
	Excludes DWord(0,1)			
	Value	Name		
	6h	Excludes Dword (0,1) [Default]		
1	31:29	Surface Type		
		Value	Name	Description
		0h	Reserved	Reserved
		1h	SURFTYPE_2D	Defines a 2-dimensional map or array of maps
		2h	Reserved	Reserved
		3h	SURFTYPE_CUBE	Defines a cube map
		4h-6h	Reserved	
		7h	SURFTYPE_NULL	Defines a null surface



3DSTATE_DEPTH_BUFFER													
	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">The Surface Type of the depth buffer must be the same as the Surface Type of the render target(s) (defined in SURFACE_STATE), unless either the depth buffer or render targets are SURFTYPE_NULL.</td> </tr> <tr> <td colspan="2">1D surface type not allowed for depth surface and stencil surface.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Workaround</th> </tr> <tr> <td colspan="2">Workaround : If depth/stencil is enabled with 1D render target, depth/stencil surface type needs to be set to 2D surface type and height set to 1. Depth will use (legacy) TileY and stencil will use TileW. For this case only, the Surface Type of the depth buffer can be 2D while the Surface Type of the render target(s) are 1D, representing an exception to a programming note above.</td> </tr> </table>	Programming Notes		The Surface Type of the depth buffer must be the same as the Surface Type of the render target(s) (defined in SURFACE_STATE), unless either the depth buffer or render targets are SURFTYPE_NULL.		1D surface type not allowed for depth surface and stencil surface.		Workaround		Workaround : If depth/stencil is enabled with 1D render target, depth/stencil surface type needs to be set to 2D surface type and height set to 1. Depth will use (legacy) TileY and stencil will use TileW. For this case only, the Surface Type of the depth buffer can be 2D while the Surface Type of the render target(s) are 1D, representing an exception to a programming note above.			
Programming Notes													
The Surface Type of the depth buffer must be the same as the Surface Type of the render target(s) (defined in SURFACE_STATE), unless either the depth buffer or render targets are SURFTYPE_NULL.													
1D surface type not allowed for depth surface and stencil surface.													
Workaround													
Workaround : If depth/stencil is enabled with 1D render target, depth/stencil surface type needs to be set to 2D surface type and height set to 1. Depth will use (legacy) TileY and stencil will use TileW. For this case only, the Surface Type of the depth buffer can be 2D while the Surface Type of the render target(s) are 1D, representing an exception to a programming note above.													
28	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Depth Write Enable</th> </tr> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">Enable</td> </tr> <tr> <td colspan="2">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.</td> </tr> </table>	Depth Write Enable		Format:	Enable	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.							
Depth Write Enable													
Format:	Enable												
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	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Stencil Write Enable</th> </tr> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">Enable</td> </tr> <tr> <td colspan="2">This field enables stencil writes to the depth buffer or stencil buffer surface, depending on where stencil is located. Both this field and the Stencil Buffer Write Enable field in DEPTH_STENCIL_STATE must be enabled in order for stencil writes to occur.</td> </tr> </table>	Stencil Write Enable		Format:	Enable	This field enables stencil writes to the depth buffer or stencil buffer surface, depending on where stencil is located. Both this field and the Stencil Buffer Write Enable field in DEPTH_STENCIL_STATE must be enabled in order for stencil writes to occur.							
Stencil Write Enable													
Format:	Enable												
This field enables stencil writes to the depth buffer or stencil buffer surface, depending on where stencil is located. Both this field and the Stencil Buffer Write Enable field in DEPTH_STENCIL_STATE must be enabled in order for stencil writes to occur.													
26:24	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Reserved</th> </tr> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Reserved		Format:	MBZ								
Reserved													
Format:	MBZ												
23	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Reserved</th> </tr> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Reserved		Format:	MBZ								
Reserved													
Format:	MBZ												
22	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Hierarchical Depth Buffer Enable</th> </tr> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">Enable</td> </tr> <tr> <td colspan="2">If enabled, indicates that a hierarchical depth buffer is defined.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If this field is enabled, the Software Tiled Rendering Mode must be NORMAL. This field must be disabled if Early Depth Test Enable is disabled OR if depth buffer surface type is NULL.</td> </tr> <tr> <td colspan="2">This field must be disabled if surface type field is SURFTYPE_1D</td> </tr> </table>	Hierarchical Depth Buffer Enable		Format:	Enable	If enabled, indicates that a hierarchical depth buffer is defined.		Programming Notes		If this field is enabled, the Software Tiled Rendering Mode must be NORMAL. This field must be disabled if Early Depth Test Enable is disabled OR if depth buffer surface type is NULL.		This field must be disabled if surface type field is SURFTYPE_1D	
Hierarchical Depth Buffer Enable													
Format:	Enable												
If enabled, indicates that a hierarchical depth buffer is defined.													
Programming Notes													
If this field is enabled, the Software Tiled Rendering Mode must be NORMAL. This field must be disabled if Early Depth Test Enable is disabled OR if depth buffer surface type is NULL.													
This field must be disabled if surface type field is SURFTYPE_1D													
21	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Reserved</th> </tr> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Reserved		Format:	MBZ								
Reserved													
Format:	MBZ												
20:18	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Surface Format</th> </tr> <tr> <td colspan="2">Specifies the format of the depth buffer. See Stencil Test Enable field in</td> </tr> </table>	Surface Format		Specifies the format of the depth buffer. See Stencil Test Enable field in									
Surface Format													
Specifies the format of the depth buffer. See Stencil Test Enable field in													



3DSTATE_DEPTH_BUFFER																			
	<p>DEPTH_STENCIL_STATE field for restrictions on the use of some of these formats.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> </tr> <tr> <td>1h</td> <td>D32_FLOAT</td> </tr> <tr> <td>2h</td> <td>Reserved</td> </tr> <tr> <td>3h</td> <td>D24_UNORM_X8_UINT</td> </tr> <tr> <td>4h</td> <td>Reserved</td> </tr> <tr> <td>5h</td> <td>D16_UNORM</td> </tr> <tr> <td>6h-7h</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	0h	Reserved	1h	D32_FLOAT	2h	Reserved	3h	D24_UNORM_X8_UINT	4h	Reserved	5h	D16_UNORM	6h-7h	Reserved		
Value	Name																		
0h	Reserved																		
1h	D32_FLOAT																		
2h	Reserved																		
3h	D24_UNORM_X8_UINT																		
4h	Reserved																		
5h	D16_UNORM																		
6h-7h	Reserved																		
17:0	<p>Surface Pitch</p> <table border="1"> <tr> <td>Format:</td> <td>U18-1 Pitch in (Bytes-1)</td> </tr> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>For TileYF and TileYS surfaces, the range is dependent on the Cu parameter (refer to <i>Memory Data Formats</i> section for the definition of the Cu parameter depending on the case). The range in bytes is $[2^{Cu}-1, 262143]$ -> $[(2^{Cu})B, 256KB] = [1 \text{ tile}, 256KB/(2^{Cu}) \text{ tiles}]$</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[7Fh, 3FFFFh]</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td> <p>The pitch specified must be a multiple of the tile pitch, in the range [128B, 128KB]. The minimum pitch should be calculated as per the formula given below.</p> <table border="1"> <thead> <tr> <th>pixel_size</th> <th>Depth Format</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Z16</td> </tr> <tr> <td>4</td> <td>Z24</td> </tr> <tr> <td>4</td> <td>Z32</td> </tr> </tbody> </table> <p>The minimum pitch should be calculated based on Cu, Cv, W_i. The Cu, Cv are the tile constants and W_i is the aligned width adjusted for MSAA. to get the Cu, Cv, W_i values and Calculations. Then use this for pitch formula : Minimum_pitch = (ceiling((W₀* pixel_size) / (1 « Cu)) * (1 « Cu)) - 1 ; //W₀ is the aligned width for the largest LOD (i.e LOD 0) (1 « Cu) = tile width in bytes (1 « Cv) = tile height in lines</p> </td> </tr> </tbody> </table>	Format:	U18-1 Pitch in (Bytes-1)	Description	For TileYF and TileYS surfaces, the range is dependent on the Cu parameter (refer to <i>Memory Data Formats</i> section for the definition of the Cu parameter depending on the case). The range in bytes is $[2^{Cu}-1, 262143]$ -> $[(2^{Cu})B, 256KB] = [1 \text{ tile}, 256KB/(2^{Cu}) \text{ tiles}]$	Value	Name	[7Fh, 3FFFFh]		Programming Notes	<p>The pitch specified must be a multiple of the tile pitch, in the range [128B, 128KB]. The minimum pitch should be calculated as per the formula given below.</p> <table border="1"> <thead> <tr> <th>pixel_size</th> <th>Depth Format</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Z16</td> </tr> <tr> <td>4</td> <td>Z24</td> </tr> <tr> <td>4</td> <td>Z32</td> </tr> </tbody> </table> <p>The minimum pitch should be calculated based on Cu, Cv, W_i. The Cu, Cv are the tile constants and W_i is the aligned width adjusted for MSAA. to get the Cu, Cv, W_i values and Calculations. Then use this for pitch formula : Minimum_pitch = (ceiling((W₀* pixel_size) / (1 « Cu)) * (1 « Cu)) - 1 ; //W₀ is the aligned width for the largest LOD (i.e LOD 0) (1 « Cu) = tile width in bytes (1 « Cv) = tile height in lines</p>	pixel_size	Depth Format	2	Z16	4	Z24	4	Z32
Format:	U18-1 Pitch in (Bytes-1)																		
Description																			
For TileYF and TileYS surfaces, the range is dependent on the Cu parameter (refer to <i>Memory Data Formats</i> section for the definition of the Cu parameter depending on the case). The range in bytes is $[2^{Cu}-1, 262143]$ -> $[(2^{Cu})B, 256KB] = [1 \text{ tile}, 256KB/(2^{Cu}) \text{ tiles}]$																			
Value	Name																		
[7Fh, 3FFFFh]																			
Programming Notes																			
<p>The pitch specified must be a multiple of the tile pitch, in the range [128B, 128KB]. The minimum pitch should be calculated as per the formula given below.</p> <table border="1"> <thead> <tr> <th>pixel_size</th> <th>Depth Format</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Z16</td> </tr> <tr> <td>4</td> <td>Z24</td> </tr> <tr> <td>4</td> <td>Z32</td> </tr> </tbody> </table> <p>The minimum pitch should be calculated based on Cu, Cv, W_i. The Cu, Cv are the tile constants and W_i is the aligned width adjusted for MSAA. to get the Cu, Cv, W_i values and Calculations. Then use this for pitch formula : Minimum_pitch = (ceiling((W₀* pixel_size) / (1 « Cu)) * (1 « Cu)) - 1 ; //W₀ is the aligned width for the largest LOD (i.e LOD 0) (1 « Cu) = tile width in bytes (1 « Cv) = tile height in lines</p>	pixel_size	Depth Format	2	Z16	4	Z24	4	Z32											
pixel_size	Depth Format																		
2	Z16																		
4	Z24																		
4	Z32																		



3DSTATE_DEPTH_BUFFER																			
<p>2..3 This field specifies the address of the buffer in mapped Graphics Memory. Programming Notes: The Depth Buffer can only be mapped to Main Memory (uncached). If the surface is tiled, the base address must conform to the Per-Surface Tiling Alignment Rules. If the buffer is linear, the surface must be 64-byte aligned.</p>	<p>63:0</p> <p>Surface Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0																
Format:	GraphicsAddress63-0																		
<p>4</p>	<p>31:18</p> <p>Height</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U14-1</td> </tr> </table> <p>This field specifies the height of the surface. If the surface is MIP-mapped, this field contains the height of the base MIP level.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> <th style="text-align: center;">Exists If</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,0]</td> <td style="text-align: center;">Legal Range</td> <td>Must be zero</td> <td>(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_1D')</td> </tr> <tr> <td style="text-align: center;">[0,16383]</td> <td style="text-align: center;">Legal Range</td> <td>Height of surface - 1 (y/v dimension)</td> <td>(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')</td> </tr> <tr> <td style="text-align: center;">[0,16383]</td> <td style="text-align: center;">Legal Range</td> <td>Height of surface - 1 (y/v dimension)</td> <td>(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_3D')</td> </tr> </tbody> </table>	Format:	U14-1	Value	Name	Description	Exists If	[0,0]	Legal Range	Must be zero	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_1D')	[0,16383]	Legal Range	Height of surface - 1 (y/v dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')	[0,16383]	Legal Range	Height of surface - 1 (y/v dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_3D')
Format:	U14-1																		
Value	Name	Description	Exists If																
[0,0]	Legal Range	Must be zero	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_1D')																
[0,16383]	Legal Range	Height of surface - 1 (y/v dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')																
[0,16383]	Legal Range	Height of surface - 1 (y/v dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_3D')																



3DSTATE_DEPTH_BUFFER			
	[0,16383]	Legal Range	y/v dimension (Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_CUBE')
Programming Notes			
The Height of the depth buffer must be the same as the Height of the render target(s) (defined in SURFACE_STATE), unless Surface Type is SURFTYPE_1D or SURFTYPE_2D with Depth = 0 (non-array) and LOD = 0 (non-mip mapped).			
17:4	Width		
	Format:	U14-1	
This field specifies the width of the surface. If the surface is MIP-mapped, this field specifies the width of the base MIP level. The width is specified in units of pixels.			
	Value	Name	Description
	[0,16383]	Legal Range	Width of surface - 1 (x/u dimension) (Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_1D')
	[0,16383]	Legal Range	Width of surface - 1 (x/u dimension) (Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')
	[0,16383]	Legal Range	Width of surface - 1 (x/u dimension) (Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_3D')
	[0,16383]	Legal Range	Width of surface - 1 (x/u dimension) (Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_CUBE')
Programming Notes			
The Width specified by this field must be less than or equal to the surface pitch (specified in bytes via the Surface Pitch field). For cube maps, Width must be set equal to Height. The Width of the depth buffer must be the same as the Width of the render target(s) (defined in SURFACE_STATE), unless Surface Type is SURFTYPE_1D or SURFTYPE_2D with Depth = 0 (non-array) and LOD = 0 (non-mip mapped).			
3:0	LOD		
	Format:	U4 for LOD units	
	Value	Name	
	[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)			



		3DSTATE_DEPTH_BUFFER	
5	31:21	Depth	
	Format:		U11-1
	<p>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.</p>		
	Value	Name	Description
	Exists If		
[0,2047]	Legal Range	Number of array elements - 1	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_1D')
[0,2047]	Legal Range	Number of array elements - 1	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')
[0,2047]	Legal Range	Depth of surface - 1 (r/z dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_3D')
[0,0]	Legal Range	Must be zero	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_CUBE')
Programming Notes			
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:		
	<p>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.</p>		
	For 3D Surfaces		
<p>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.</p>			
For Other Surfaces			
This field is ignored			
[0,2047]	SURFTYPE_1D/2D	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_1D' Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')	
[0,2047]	SURFTYPE_3D	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_3D')	
9:7	Reserved		
Format:		MBZ	
6:0	Depth Buffer Object Control State		
Format:		MEMORY_OBJECT_CONTROL_STATE	
Specifies the memory object control state for the depth buffer.			



3DSTATE_DEPTH_BUFFER																																																								
6	31:30	<p>Tiled Resource Mode For Depth Buffer Surfaces: This field specifies the tiled resource mode. For other surfaces: This field is ignored.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">NONE</td> <td>No tiled resource</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td style="text-align: center;">2h</td> <td style="text-align: center;">TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td style="text-align: center;">3h</td> <td style="text-align: center;">Reserved</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>If Tile Mode is not set to TILEMODE_YMAJOR, this field must be set to TRMODE_NONE. HIZ and Stencil surfaces can't be tiled resource, hence will continue to follow legacy tiledY and tiledW respectively.</p>	Value	Name	Description	0h	NONE	No tiled resource	1h	TILEYF	4KB tiled resources	2h	TILEYS	64KB tiled resources	3h	Reserved																																								
Value	Name	Description																																																						
0h	NONE	No tiled resource																																																						
1h	TILEYF	4KB tiled resources																																																						
2h	TILEYS	64KB tiled resources																																																						
3h	Reserved																																																							
	29:26	<p>Mip Tail Start LOD Format: U4 in LOD Units</p> <p>For Sampling Engine, Render Target, and Typed Surfaces: This field indicates which LOD is the first one in the MIP tail if Tiled Mode is not TRMODE_NONE. The MIP tail has a different layout than the rest of the surface. Refer to the <i>Memory Data Formats</i> section for more details. For other surfaces: This field is ignored.</p> <p style="text-align: center;">Programming Notes</p> <p>This field must be zero if the Surface Format is MONO8 This field is ignored if Tiled Mode is TRMODE_NONE unless Surface Type is SURFTYPE_1D. If Tiled Mode is not TRMODE_NONE, this field must be set to ensure that mips within the mip tail do not overlap given the storage algorithms given in the Memory Data Formats section. The following table indicates the <i>maximum</i> size of the mip that is set to be the Mip Tail Start LOD for various cases:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th rowspan="2" style="text-align: center;">Surface Type</th> <th rowspan="2" style="text-align: center;">Tiling Mode</th> <th rowspan="2" style="text-align: center;">#MS</th> <th colspan="5" style="text-align: center;">Bits Per Element</th> </tr> <tr> <th style="text-align: center;">8</th> <th style="text-align: center;">16</th> <th style="text-align: center;">32</th> <th style="text-align: center;">64</th> <th style="text-align: center;">128</th> </tr> </thead> <tbody> <tr> <td rowspan="2" style="text-align: center;">1D</td> <td style="text-align: center;">64KB</td> <td style="text-align: center;">1</td> <td style="text-align: center;">16384</td> <td style="text-align: center;">8192</td> <td style="text-align: center;">4096</td> <td style="text-align: center;">2048</td> <td style="text-align: center;">1024</td> </tr> <tr> <td style="text-align: center;">4KB</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1024</td> <td style="text-align: center;">512</td> <td style="text-align: center;">256</td> <td style="text-align: center;">128</td> <td style="text-align: center;">64</td> </tr> <tr> <td rowspan="4" style="text-align: center;">2D/ CUBE</td> <td rowspan="4" style="text-align: center;">TileYS</td> <td style="text-align: center;">1</td> <td style="text-align: center;">128x256</td> <td style="text-align: center;">128x128</td> <td style="text-align: center;">64x128</td> <td style="text-align: center;">64x64</td> <td style="text-align: center;">32x64</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">128x128</td> <td style="text-align: center;">64x128</td> <td style="text-align: center;">64x64</td> <td style="text-align: center;">32x64</td> <td style="text-align: center;">32x32</td> </tr> <tr> <td style="text-align: center;">4</td> <td style="text-align: center;">64x128</td> <td style="text-align: center;">64x64</td> <td style="text-align: center;">32x64</td> <td style="text-align: center;">32x32</td> <td style="text-align: center;">16x32</td> </tr> <tr> <td style="text-align: center;">8</td> <td style="text-align: center;">64x64</td> <td style="text-align: center;">32x64</td> <td style="text-align: center;">32x32</td> <td style="text-align: center;">16x32</td> <td style="text-align: center;">16x16</td> </tr> </tbody> </table>	Surface Type	Tiling Mode	#MS	Bits Per Element					8	16	32	64	128	1D	64KB	1	16384	8192	4096	2048	1024	4KB	1	1024	512	256	128	64	2D/ CUBE	TileYS	1	128x256	128x128	64x128	64x64	32x64	2	128x128	64x128	64x64	32x64	32x32	4	64x128	64x64	32x64	32x32	16x32	8	64x64	32x64	32x32	16x32	16x16
Surface Type	Tiling Mode	#MS				Bits Per Element																																																		
			8	16	32	64	128																																																	
1D	64KB	1	16384	8192	4096	2048	1024																																																	
	4KB	1	1024	512	256	128	64																																																	
2D/ CUBE	TileYS	1	128x256	128x128	64x128	64x64	32x64																																																	
		2	128x128	64x128	64x64	32x64	32x32																																																	
		4	64x128	64x64	32x64	32x32	16x32																																																	
		8	64x64	32x64	32x32	16x32	16x16																																																	



3DSTATE_DEPTH_BUFFER									
			16	32x64	32x32	16x32	16x16	8x16	
		TileYF	any	64x32	32x32	32x16	16x16	16x8	
		3D	TileYS	1	32x32x32	16x32x32	16x32x16	16x16x16	8x16x16
			TileYF	1	16x8x16	8x8x16	8x8x8	8x4x8	4x4x8
	25:0	Reserved							
		Format:					MBZ		
7	31:21	Render Target View Extent							
		Format:					U11-1		
		Value	Name	Description	Exists If				
		[0,2047]	Legal Range	Number of array elements- 1	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_1D')				
		[0,2047]	Legal Range	Number of array elements- 1	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')				
		[0,2047]	Legal Range	To indication extent of [1,2048]	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_3D')				
		[0,0]	Legal Range	Must be zero	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_CUBE')				
	20:15	Reserved							
	14:0	Surface QPitch							
		Format:					QPitch[16:2]		
		Description							
		The interpretation of this field is dependent on Surface Type as follows:							
		<ul style="list-style-type: none"> • SURFTYPE_1D: distance in <i>pixels</i> between array slices • SURFTYPE_2D/CUBE: distance in <i>rows</i> between array slices • SURFTYPE_3D: distance in <i>rows</i> between R-slices 							
		For 1D and 2D Surfaces:							
		This field must be set to the same value as the Depth field.							
		For 3D Surfaces:							
		This field indicates the extent of the accessible 'R' coordinates minus 1 on the LOD currently being rendered to.							
		For Other Surfaces							
		This field is ignored.							
		Value	Name	Description					
		[4h, 1FFFCh]		in multiples of 4 (low 2 bits missing)					



3DSTATE_DEPTH_BUFFER

Programming Notes

Software must ensure that this field is set to a value sufficiently large that array slices in the surface do not overlap. Refer to the *Memory Data Formats* section for information on how surfaces are stored.

TYS/TYF QPitch is valid only for 2D array surfaces and represents the tile-padded total number of texels(lines) in a single array slice.

Height of each LOD:

$HL = \text{AlignToTileHeight}(\text{MSAA_height_factor} * (\mathbf{height} \gg L) > 0 ? \mathbf{height} \gg L : 1)$, where $\text{AlignToTileHeight}(x)$ is $(\text{ceiling}(x) / (1 \ll Cv)) * (1 \ll Cv)$

Height of all LODs is a sum:

$H = H_0 + H_1 + \dots + H_n$,

N is number of mip levels.

If surface has MIP tail, equation stops at H_n where $n = \text{MipTailStartLOD}$. MipTail is single tile.

QPitch is multiple of tile height $(1 \ll Cv)$ and should be equal or greater H computed above.



3DSTATE_DRAWING_RECTANGLE

3DSTATE_DRAWING_RECTANGLE			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_DRAWING_RECTANGLE command is used to set the 3D drawing rectangle and related state.			
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:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		00h 3DSTATE_DRAWING_RECTANGLE	
Format:		OpCode	
15:14	Core Mode Select		
	Format:	U2	
	Specifies which core this command will be considered valid and update based on the state in this command.		
	Value	Name	Description
	0h	Legacy	Both cores are enabled and will update the state.
	1h	Core 0 Enabled	State will be updated in Core 0 only
	2h	Core 1 Enabled	State will be updated in Core 1 only
3h	Reserved		
13:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	2h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:16	Clipped Drawing Rectangle Y Min	
		Format:	U16 in Pixels from Color Buffer origin (upper left corner)
Specifies Ymin value of (inclusive) intersection of Drawing rectangle with the Color (Destination) Buffer, used for clipping. Pixels with Y coordinates less than Ymin will be clipped out.			



3DSTATE_DRAWING_RECTANGLE																	
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Device ignores bits 31:30</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This value can be larger than Clipped Drawing Rectangle Y Max. If Ymin>Ymax, the clipped drawing rectangle is null, all polygons are discarded. If Ymin==Ymax, the clipped drawing rectangle is 1 pixel wide in the Y direction.</p>	Value	Name	[0,16383]	Device ignores bits 31:30												
Value	Name																
[0,16383]	Device ignores bits 31:30																
15:0	<p>Clipped Drawing Rectangle X Min</p> <table border="1"> <tr> <td>Format:</td> <td>U16 in Pixels from Color Buffer origin (upper left corner)</td> </tr> </table> <p>Specifies Xmin value of (inclusive) intersection of Drawing rectangle with the Color (Destination) Buffer, used for clipping. Pixels with X coordinates less than Xmin will be clipped out.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Device ignores bits 15:14</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This value can be larger than Clipped Drawing Rectangle X Max. If Xmin>Xmax, the clipped drawing rectangle is null, all polygons are discarded. If Xmin==Xmax, the clipped drawing rectangle is 1 pixel wide in the X direction.</p>	Format:	U16 in Pixels from Color Buffer origin (upper left corner)	Value	Name	[0,16383]	Device ignores bits 15:14										
Format:	U16 in Pixels from Color Buffer origin (upper left corner)																
Value	Name																
[0,16383]	Device ignores bits 15:14																
2	<table border="1"> <tr> <td>31:16</td> <td> <p>Clipped Drawing Rectangle Y Max</p> <table border="1"> <tr> <td>Format:</td> <td>U16 in Pixels from Color Buffer origin (upper left corner)</td> </tr> </table> <p>Specifies Ymax value of (inclusive) intersection of Drawing rectangle with the Color (Destination) Buffer, used for clipping. Pixels with coordinates greater than Ymax will be clipped out.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Device ignores bits 31:30</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This value can be less than Clipped Drawing Rectangle Y Min. If Ymax<Ymin, the clipped drawing rectangle is null, all polygons are discarded. If Ymin==Ymax, the clipped drawing rectangle is 1 pixel wide in the Y direction.</p> </td> </tr> <tr> <td>15:0</td> <td> <p>Clipped Drawing Rectangle X Max</p> <table border="1"> <tr> <td>Format:</td> <td>U16 in Pixels from Color Buffer origin (upper left corner)</td> </tr> </table> <p>Specifies Xmax value of (inclusive) intersection of Drawing rectangle with the Color (Destination) Buffer, used for clipping. Pixels with coordinates greater than Xmax will be clipped out.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Device ignores bits 15:14</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This value can be less than Clipped Drawing Rectangle X Min. If Xmax<Xmin, the clipped drawing rectangle is null, all polygons are discarded. If Xmin==Xmax, the clipped drawing</p> </td> </tr> </table>	31:16	<p>Clipped Drawing Rectangle Y Max</p> <table border="1"> <tr> <td>Format:</td> <td>U16 in Pixels from Color Buffer origin (upper left corner)</td> </tr> </table> <p>Specifies Ymax value of (inclusive) intersection of Drawing rectangle with the Color (Destination) Buffer, used for clipping. Pixels with coordinates greater than Ymax will be clipped out.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Device ignores bits 31:30</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This value can be less than Clipped Drawing Rectangle Y Min. If Ymax<Ymin, the clipped drawing rectangle is null, all polygons are discarded. If Ymin==Ymax, the clipped drawing rectangle is 1 pixel wide in the Y direction.</p>	Format:	U16 in Pixels from Color Buffer origin (upper left corner)	Value	Name	[0,16383]	Device ignores bits 31:30	15:0	<p>Clipped Drawing Rectangle X Max</p> <table border="1"> <tr> <td>Format:</td> <td>U16 in Pixels from Color Buffer origin (upper left corner)</td> </tr> </table> <p>Specifies Xmax value of (inclusive) intersection of Drawing rectangle with the Color (Destination) Buffer, used for clipping. Pixels with coordinates greater than Xmax will be clipped out.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Device ignores bits 15:14</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This value can be less than Clipped Drawing Rectangle X Min. If Xmax<Xmin, the clipped drawing rectangle is null, all polygons are discarded. If Xmin==Xmax, the clipped drawing</p>	Format:	U16 in Pixels from Color Buffer origin (upper left corner)	Value	Name	[0,16383]	Device ignores bits 15:14
31:16	<p>Clipped Drawing Rectangle Y Max</p> <table border="1"> <tr> <td>Format:</td> <td>U16 in Pixels from Color Buffer origin (upper left corner)</td> </tr> </table> <p>Specifies Ymax value of (inclusive) intersection of Drawing rectangle with the Color (Destination) Buffer, used for clipping. Pixels with coordinates greater than Ymax will be clipped out.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Device ignores bits 31:30</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This value can be less than Clipped Drawing Rectangle Y Min. If Ymax<Ymin, the clipped drawing rectangle is null, all polygons are discarded. If Ymin==Ymax, the clipped drawing rectangle is 1 pixel wide in the Y direction.</p>	Format:	U16 in Pixels from Color Buffer origin (upper left corner)	Value	Name	[0,16383]	Device ignores bits 31:30										
Format:	U16 in Pixels from Color Buffer origin (upper left corner)																
Value	Name																
[0,16383]	Device ignores bits 31:30																
15:0	<p>Clipped Drawing Rectangle X Max</p> <table border="1"> <tr> <td>Format:</td> <td>U16 in Pixels from Color Buffer origin (upper left corner)</td> </tr> </table> <p>Specifies Xmax value of (inclusive) intersection of Drawing rectangle with the Color (Destination) Buffer, used for clipping. Pixels with coordinates greater than Xmax will be clipped out.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Device ignores bits 15:14</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This value can be less than Clipped Drawing Rectangle X Min. If Xmax<Xmin, the clipped drawing rectangle is null, all polygons are discarded. If Xmin==Xmax, the clipped drawing</p>	Format:	U16 in Pixels from Color Buffer origin (upper left corner)	Value	Name	[0,16383]	Device ignores bits 15:14										
Format:	U16 in Pixels from Color Buffer origin (upper left corner)																
Value	Name																
[0,16383]	Device ignores bits 15:14																



3DSTATE_DRAWING_RECTANGLE		
		rectangle is 1 pixel wide in the X direction.
3	31:16	<p>Drawing Rectangle Origin Y</p> <p>Format: S15 in Pixels from Color Buffer origin (upper left corner).</p> <p style="text-align: center;">Description</p> <p>Range: [-16384,16383] (Bit 31 should be a sign extension)</p> <p>Specifies Y 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.</p>
	15:0	<p>Drawing Rectangle Origin X</p> <p>Format: S15 in Pixels from Color Buffer origin (upper left corner).</p> <p style="text-align: center;">Description</p> <p>Range: [-16384,16383] (Bit 15 should be a sign extension)</p> <p>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.</p>



3DSTATE_DS

3DSTATE_DS		
Source:		RenderCS
Length Bias:		2
The state used by DS is defined with this inline state packet		
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: 0h 3DSTATE_PIPELINED Format: OpCode
	23:16	3D Command Sub Opcode
Default Value: 1Dh 3DSTATE_DS Format: OpCode		
15:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Default Value: 9h Excludes DWord (0,1) Format: =n Total Length - 2	
1..2	63:6	Kernel Start Pointer
		Format: InstructionBaseOffset[63:6]Kernel This field specifies the starting location of the kernel program run by threads spawned by this FF unit. 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
		Format: MBZ
3	31	Reserved
		Format: MBZ



3DSTATE_DS																						
30	<p>Vector Mask Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U1 Enumerated Type</td> </tr> </table> <p>Upon subsequent DS thread dispatches, this bit is loaded into the EU's Vector Mask Enable (VME, cr0.0[3]) thread state. Refer to EU documentation for the definition and use of VME state.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Dmask</td> <td>The EU will use the Dispatch Mask (supplied by the DS stage) for instruction execution.</td> </tr> <tr> <td>1h</td> <td>Vmask</td> <td>The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Under normal conditions SW shall specify DMask, as the DS stage will provide a Dispatch Mask appropriate to SIMD4x2 or SIMD8 thread execution (as a function of dispatch mode). E.g., for SIMD4x2 thread execution, the DS stage will generate a Dispatch Mask that is equal to what the EU would use as the Vector Mask. For SIMD8 execution there is no known usage model for use of Vector Mask (as there is for PS shaders).</p>		Format:	U1 Enumerated Type	Value	Name	Description	0h	Dmask	The EU will use the Dispatch Mask (supplied by the DS stage) for instruction execution.	1h	Vmask	The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.									
Format:	U1 Enumerated Type																					
Value	Name	Description																				
0h	Dmask	The EU will use the Dispatch Mask (supplied by the DS stage) for instruction execution.																				
1h	Vmask	The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.																				
29:27	<p>Sampler Count</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U3</td> </tr> </table> <p>Specifies how many samplers (in multiples of 4) the kernel uses. Used only for prefetching the associated sampler state entries. This field is ignored if DS Function Enable is DISABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No Samplers</td> <td>No samplers used</td> </tr> <tr> <td>1h</td> <td>1-4 Samplers</td> <td>between 1 and 4 samplers used</td> </tr> <tr> <td>2h</td> <td>5-8 Samplers</td> <td>between 5 and 8 samplers used</td> </tr> <tr> <td>3h</td> <td>9-12 Samplers</td> <td>between 9 and 12 samplers used</td> </tr> <tr> <td>4h</td> <td>13-16 Samplers</td> <td>between 13 and 16 samplers used</td> </tr> </tbody> </table>		Format:	U3	Value	Name	Description	0h	No Samplers	No samplers used	1h	1-4 Samplers	between 1 and 4 samplers used	2h	5-8 Samplers	between 5 and 8 samplers used	3h	9-12 Samplers	between 9 and 12 samplers used	4h	13-16 Samplers	between 13 and 16 samplers used
Format:	U3																					
Value	Name	Description																				
0h	No Samplers	No samplers used																				
1h	1-4 Samplers	between 1 and 4 samplers used																				
2h	5-8 Samplers	between 5 and 8 samplers used																				
3h	9-12 Samplers	between 9 and 12 samplers used																				
4h	13-16 Samplers	between 13 and 16 samplers used																				
26	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ																		
Format:	MBZ																					
25:18	<p>Binding Table Entry Count</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U8</td> </tr> </table> <p>When HW Generated Binding Table is disabled: Specifies how many binding table entries the kernel uses. Used only for prefetching of the binding table entries and associated surface state. Note: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. This field is ignored if DS Function Enable is DISABLED.</p> <p>When HW Generated Binding Table bit is enabled:</p>		Format:	U8																		
Format:	U8																					



3DSTATE_DS											
	<p>This field indicates which cache lines (512bit units - 32 Binding Table Entry section) should be fetched. Each bit in this field corresponds to a cache line. Only the 1st 4 non-zero Binding Table entries of each 32 Binding Table entry section prefetched will have its surface state prefetched.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,255]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.</p>		Value	Name	[0,255]						
Value	Name										
[0,255]											
17	<p>Thread Dispatch Priority</p> <p>Format: U1 Enumerated Type</p> <p>Specifies the priority of the thread for dispatch: This field is ignored if DS Function Enable is DISABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%; text-align: center;">Value</th> <th style="width: 25%; text-align: center;">Name</th> <th style="width: 50%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">Normal</td> <td style="text-align: center;">Normal Priority</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">High</td> <td style="text-align: center;">High Priority</td> </tr> </tbody> </table>		Value	Name	Description	0h	Normal	Normal Priority	1h	High	High Priority
Value	Name	Description									
0h	Normal	Normal Priority									
1h	High	High Priority									
16	<p>Floating Point Mode</p> <p>Format: U1 Enumerated Type</p> <p>Specifies the initial floating point mode used by the dispatched thread. This field is ignored if DS Function Enable is DISABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%; text-align: center;">Value</th> <th style="width: 25%; text-align: center;">Name</th> <th style="width: 50%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">IEEE-754</td> <td style="text-align: center;">Use IEEE-754 Rules</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">Alternate</td> <td style="text-align: center;">Use alternate rules</td> </tr> </tbody> </table>		Value	Name	Description	0h	IEEE-754	Use IEEE-754 Rules	1h	Alternate	Use alternate rules
Value	Name	Description									
0h	IEEE-754	Use IEEE-754 Rules									
1h	Alternate	Use alternate rules									
15	<p>Reserved</p> <p>Format: MBZ</p>										
14	<p>Accesses UAV</p> <p>Format: Enable</p> <p>This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.</p> <p style="text-align: center;">Programming Notes</p> <p>This field must not be set when DS Function Enable is disabled.</p>										
13	<p>Illegal Opcode Exception Enable</p> <p>Format: Enable</p> <p>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.</p>										
12:8	<p>Reserved</p>										



		3DSTATE_DS				
		Format:	MBZ			
	7	Software Exception Enable Format: Enable This bit gets loaded 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				
4..5	63:10	Scratch Space Base Pointer Format: GeneralStateOffset[63:10]ScratchSpace 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 Format: MBZ				
	3:0	Per-Thread Scratch Space 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 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.				
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,11]</td> <td>indicating [1K Bytes, 2M Bytes]</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> 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.		Value	Name	[0,11]
Value	Name					
[0,11]	indicating [1K Bytes, 2M Bytes]					
6	31:25	Reserved Format: MBZ				
	24:20	Dispatch GRF Start Register For URB Data Format: GRFRegister[4:0] Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload.				



3DSTATE_DS								
		<p>This field is ignored if DS Function Enable is DISABLED.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,31]</td> <td></td> <td>indicating GRF [R0, R31]</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>When SIMD8_SINGLE_OR_DUAL_PATCH dispatch mode is selected, SW shall program this field to a value of 6 or greater to accommodate either the DUAL_PATCH or SIMD8_SINGLE_PATCH payloads.</p>	Value	Name	Description	[0,31]		indicating GRF [R0, R31]
Value	Name	Description						
[0,31]		indicating GRF [R0, R31]						
	19:18	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ							
	17:11	<p>Patch URB Entry Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U7</td> </tr> </table> <p>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.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,64]</td> <td></td> </tr> </tbody> </table>	Format:	U7	Value	Name	[0,64]	
Format:	U7							
Value	Name							
[0,64]								
	10	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ							
	9:4	<p>Patch URB Entry Read Offset</p> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>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.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,63]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,63]	
Format:	U6							
Value	Name							
[0,63]								
	3:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ							
7	31	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ						
	30	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ							
29:21	<p>Maximum Number of Threads</p> <table border="1"> <tr> <td>Format:</td> <td>U9-1 Thread Count</td> </tr> </table> <p>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</p>	Format:	U9-1 Thread Count					
Format:	U9-1 Thread Count							



3DSTATE_DS			
		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
		[0,335]	indicating thread count of [1,336]
20:11	Reserved	Format:	MBZ
10	Statistics Enable	Format:	Enable
	If ENABLED, this FF unit will engage in statistics gathering. Refer to the Statistics Gathering section. If DISABLED, statistics information associated with this FF stage will be left unchanged. This field is ignored if DS Function Enable is DISABLED.		
9:5	Reserved	Format:	MBZ
4:3	Dispatch Mode	Format:	U2
	This field specifies how the DS stage generates DS thread requests, and correspondingly impacts the DS thread payload. The setting of this field must agree with how the DS kernel was compiled. This field is ignored if DS Function Enable is DISABLED.		
	Value	Name	Description
	Programming Notes		
0h	SIMD4X2	DS threads are passed one patch, up to 2 domain point inputs, and up to two output vertex handles. The DS kernel (at KSP) is expected to run in SIMD4x2 execution mode. The DUAL_PATCH KSP is ignored. The Single Domain Point Dispatch field can be used to force single domain point dispatches.	
1h	SIMD8_SINGLE_PATCH	DS threads are passed one patch, up to 8 domain point inputs, and up to 8 output vertex handles. All the domain points are associated with the single input patch. The DS kernel (at KSP) is expected to run in SIMD8 execution mode. The DUAL_PATCH KSP is ignored.	



3DSTATE_DS						
	2h	SIMD8_SINGLE_OR_DUAL_PATCH	<p>SIMD8_SINGLE_OR_DUAL_PATCH This mode enables use of both the KSP and the DUAL_PATCH KSP. The KSP kernel operates just like in SIMD8_SINGLE_PATCH mode. See DUAL_PATCH Thread Execution for a discussion of how the DUAL_PATCH KSP is used.</p> <p>At least 2 HS URB handles must be allocated in order to enable this mode.</p> <p>SIMD8_SINGLE_OR_DUAL_PATCH must not be used if the domain shader kernel uses primitive id.</p>			
	3h	Reserved				
	2	<p>Compute W Coordinate Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>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.</p> <p>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.</p> <p>This field is ignored if DS Function Enable is DISABLED.</p>		Format:	Enable	
Format:	Enable					
	1	<p>Cache Disable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Disable</td> </tr> </table> <p>This bit controls the operation of the DS Cache. This field is ignored if DS Function Enable is DISABLED.</p> <p>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.</p> <p>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.</p> <p>The DS Cache is invalidated whenever the DS Cache becomes DISABLED, whenever the DS Function Enable toggles, and between patches.</p>		Format:	Disable	
Format:	Disable					
	0	<p>Function Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, DS threads will be spawned to process incoming domain points which miss in the DS cache.</p> <p>If DISABLED, the DS stage goes into pass-through mode and performs no specific processing. This field is always used.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>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.</p>		Format:	Enable	Programming Notes
Format:	Enable					
Programming Notes						
8	31:28	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ	
Format:	MBZ					
	27	<p>Reserved</p>				



3DSTATE_DS					
26:21	Vertex URB Entry Output Read Offset				
	Format: U6				
	Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB by SBE.				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,63]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,63]	
Value	Name				
[0,63]					
20:16	Vertex URB Entry Output Length				
	Format: U5				
	Specifies the amount of URB data written for each Vertex URB entry, in 256-bit register increments.				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[1,16]</td> <td></td> </tr> </tbody> </table>	Value	Name	[1,16]	
	Value	Name			
[1,16]					
Programming Notes					
This length does not include the vertex header.					
15:8	User Clip Distance Clip Test Enable Bitmask				
	Format: U8 This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept / must clip determination needs to be made. DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.				
7:0	User Clip Distance Cull Test Enable Bitmask				
	Format: U8 This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept determination needs to be made (does not cause a must clip). DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.				
9..10	63:6 DUAL_PATCH Kernel Start Pointer				
	Format: InstructionBaseOffset[63:6]Kernel This field specifies the starting location of the DUAL_PATCH kernel program run by threads spawned by this FF unit. It is specified as a 64-byte-granular offset from the Instruction Base Address. This field is ignored if DS Function Enable is DISABLED. See DUAL_PATCH Thread Execution for a discussion of how the DUAL_PATCH KSP is used.				
5:0	Reserved				
	Format: MBZ				



3DSTATE_DX9_CONSTANT_BUFFER_POOL_ALLOC

3DSTATE_DX9_CONSTANT_BUFFER_POOL_ALLOC			
Source:	RenderCS		
Length Bias:	2		
This command sets up the Gather Pool for Gather Buffers.			
Programming Notes			
This command must only be programmed when resource streamer is enabled thru batch buffer start and MI_RS_CONTROL has not disabled resource streamer.			
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
	GFXPIPE_3D		
	26:24	3D Command Opcode	
		Default Value:	1h 3DSTATE_NONPIPELINED
		Format:	OpCode
23:16	3D Command Sub Opcode		
	Default Value:	1Bh 3DSTATE_DX9_CONSTANT_BUFFER_POOL_ALLOC	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Format:	=n	
	Total Length - 2		
	Value	Name	
2h	DWORD_COUNT_n [Default]		
1..2	63:48	Reserved	
		Format:	MBZ



3DSTATE_DX9_CONSTANT_BUFFER_POOL_ALLOC			
47:13	<p>Dx9 Constant Buffer Pool Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>GraphicsAddress[47:13]Dx9_Constant_Buffer_Pool</td> </tr> </table> <p>This field specifies the 4K-byte aligned (exclusive) maximum Graphics Memory address access by an 3DSTATE_CONSTANT_BUFFER_* command when HW generate Dx9 Constant Buffers are enabled. Indirect data accessed at this address and beyond will appear to be 0. Setting this field to 0 will cause this range check to be ignored.</p>	Format:	GraphicsAddress[47:13]Dx9_Constant_Buffer_Pool
	Format:	GraphicsAddress[47:13]Dx9_Constant_Buffer_Pool	
	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ	
	<p>Dx9 Constant Buffer Pool Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>Enable</td> </tr> </table> <p>When this bit is set it enables HW Dx9 constants buffers. When this bit is cleared it disables HW Dx9 constant buffers, the local bits for the constant buffers are cleared and the buffers will not be save or restored as part of context.</p>	Format:	Enable
	Format:	Enable	
<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ		
<p>Surface Object Control State</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for this surface.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>Bit 2 is not programmable and is always zero.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE	Programming Notes
Format:	MEMORY_OBJECT_CONTROL_STATE		
Programming Notes			
<p>Dx9 Constant Buffer Pool Buffer Size</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>U19</td> </tr> </table> <p>This field specifies the size of the buffer in 8K pages. Any access which straddle or go past the end of the buffer will return 0. Note that BufferSize=0 indicates that there is no valid data in the buffer.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Restriction</td> </tr> </table> <p>Restriction : Programming size of zero is illegal in the case that the pool is enabled.</p>	Format:	U19	Restriction
Format:	U19		
Restriction			
<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ		



3DSTATE_DX9_CONSTANTB_PS

3DSTATE_DX9_CONSTANTB_PS			
Source:	RenderCS		
Length Bias:	2		
This command sets a DX9 constant Boolean register for PS.			
Programming Notes			
<ul style="list-style-type: none"> The 3DSTATE_DX9_CONSTANTB_PS is a variable length command. Programming this command in batch buffer requires that all float, integer and boolean constants initialized prior to any commands or events that cause the constants to be written to memory. 			
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:	0h GFXPIPE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		3Eh 3DSTATE_DX9_CONSTANTB_PS	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Format:	=n Total Length - 2	
	Value	Name	
	0h	[Default]	
	0h-10h	Excludes DWord (0,1)	
1	31:16	Reserved	
		Format:	MBZ



3DSTATE_DX9_CONSTANTB_PS		
	15	Global Constant Register Format: U1 When this bit is set the global constant register set will be updated. When this bit is clear the local constant register set will be updated.
	14:4	Reserved Format: MBZ
	3:0	Constant Register Index Format: U4 This field specifies the index of 1st boolean to be updated.
2..n	31:0	Entry Format: DX9_CONSTANTB_ENTRY The nth boolean to be updated.



3DSTATE_DX9_CONSTANTB_VS

3DSTATE_DX9_CONSTANTB_VS			
Source:	RenderCS		
Length Bias:	2		
This command sets a DX9 constant Boolean register for PS.			
Programming Notes			
<ul style="list-style-type: none"> The 3DSTATE_DX9_CONSTANTB_VS is a variable length command. Programming this command in batch buffer requires that all float, integer and boolean constants initialized prior to any commands or events that cause the constants to be written to memory. 			
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:	0h GFXPIPE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		3Dh 3DSTATE_DX9_CONSTANTB_VS	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Format:	=n Total Length - 2	
	Value	Name	
	0h	[Default]	
	0h-10h	Excludes DWord (0,1)	
1	31:16	Reserved	
		Format:	MBZ



3DSTATE_DX9_CONSTANTB_VS				
	15	Global Constant Register <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> <p>When this bit is set the global constant register set will be updated. When this bit is clear the local constant register set will be updated.</p>	Format:	U1
	Format:	U1		
	14:4	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
3:0	Constant Register Index <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>This field specifies the index of 1st boolean to be updated.</p>	Format:	U4	
Format:	U4			
2..n	31:0	Entry <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">DX9_CONSTANTB_ENTRY</td> </tr> </table> <p>The nth boolean to be updated.</p>	Format:	DX9_CONSTANTB_ENTRY
Format:	DX9_CONSTANTB_ENTRY			



3DSTATE_DX9_CONSTANTF_PS

3DSTATE_DX9_CONSTANTF_PS							
Source:	RenderCS						
Length Bias:	2						
This command sets one or more DX9 constant float registers for PS.							
Programming Notes							
<ul style="list-style-type: none"> The 3DSTATE_DX9_CONSTANTF_PS is a variable length command. Programming this command in batch buffer requires that all float, integer and boolean constants initialized prior to any commands or events that cause the constants to be written to memory. 							
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: 0h GFXPIPE_PIPELINED Format: OpCode					
	23:16	3D Command Sub Opcode					
Default Value: 3Ah 3DSTATE_DX9_CONSTANTF_PS Format: OpCode							
15:11	Reserved						
	Format: MBZ						
10:0	DWord Length						
	Format: =n Total Length - 2						
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> <tr> <td>1h-400h</td> <td>multiples of 4</td> </tr> </tbody> </table>	Value	Name	1h	Excludes DWord (0,1) [Default]	1h-400h	multiples of 4
	Value	Name					
1h	Excludes DWord (0,1) [Default]						
1h-400h	multiples of 4						
1	31:16	Reserved					
		Format: MBZ					



3DSTATE_DX9_CONSTANTF_PS				
	15	<p>Global Constant Register</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> <p>When this bit is set the global constant register set will be updated. When this bit is clear the local constant register set will be updated.</p>	Format:	U1
	Format:	U1		
	14:8	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
7:0	<p>Constant Register Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U8</td> </tr> </table> <p>This field specifies the index of 1st 4 component float to be updated.</p>	Format:	U8	
Format:	U8			
2..n	127:0	<p>Entry</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">DX9_CONSTANTF_ENTRY</td> </tr> </table> <p>The four components of the nth float to be updated.</p>	Format:	DX9_CONSTANTF_ENTRY
Format:	DX9_CONSTANTF_ENTRY			



3DSTATE_DX9_CONSTANTF_VS

3DSTATE_DX9_CONSTANTF_VS			
Source:	RenderCS		
Length Bias:	2		
This command sets one or more DX9 constant float registers for VS.			
Programming Notes			
<ul style="list-style-type: none"> The 3DSTATE_DX9_CONSTANTF_VS is a variable length command. Programming this command in batch buffer requires that all float, integer and boolean constants initialized prior to any commands or events that cause the constants to be written to memory. 			
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:	0h GFXPIPE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		39h 3DSTATE_DX9_CONSTANTF_VS	
Format:		OpCode	
15:11	Reserved		
	Format:	MBZ	
10:0	DWord Length		
	Format:	=n Total Length - 2	
	Value	Name	
	1h	Excludes DWord (0,1) [Default]	
1h-400h	multiples of 4		
1	31:16	Reserved	
		Format:	MBZ



3DSTATE_DX9_CONSTANTF_VS				
	15	<p>Global Constant Register</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> <p>When this bit is set the global constant register set will be updated. When this bit is clear the local constant register set will be updated.</p>	Format:	U1
	Format:	U1		
	14:8	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
7:0	<p>Constant Register Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U8</td> </tr> </table> <p>This field specifies the index of 1st 4 component float to be updated.</p>	Format:	U8	
Format:	U8			
2..n	127:0	<p>Entry</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">DX9_CONSTANTF_ENTRY</td> </tr> </table> <p>The four components of the nth float to be updated.</p>	Format:	DX9_CONSTANTF_ENTRY
Format:	DX9_CONSTANTF_ENTRY			



3DSTATE_DX9_CONSTANTI_PS

3DSTATE_DX9_CONSTANTI_PS										
Source:	RenderCS									
Length Bias:	2									
This command sets one or more DX9 constant integer registers for PS.										
Programming Notes										
<ul style="list-style-type: none"> The 3DSTATE_DX9_CONSTANTI_PS is a variable length command. Programming this command in batch buffer requires that all float, integer and boolean constants initialized prior to any commands or events that cause the constants to be written to memory. 										
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: 0h GFXPIPE_PIPELINED Format: OpCode								
	23:16	3D Command Sub Opcode								
Default Value: 3Ch 3DSTATE_DX9_CONSTANTI_PS Format: OpCode										
15:8	Reserved									
	Format: MBZ									
7:0	DWord Length									
	Format: =n Total Length - 2									
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td>[Default]</td> <td>Excludes DWord (0,1)</td> </tr> <tr> <td>0h-80h</td> <td>multiples of 4</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	1h	[Default]	Excludes DWord (0,1)	0h-80h	multiples of 4	
	Value	Name	Description							
1h	[Default]	Excludes DWord (0,1)								
0h-80h	multiples of 4									
1	31:16	Reserved								
		Format: MBZ								



3DSTATE_DX9_CONSTANTI_PS				
	15	<p>Global Constant Register</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> <p>When this bit is set the global constant register set will be updated. When this bit is clear the local constant register set will be updated.</p>	Format:	U1
	Format:	U1		
	14:5	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
4:0	<p>Constant Register Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U5</td> </tr> </table> <p>This field specifies the index of 1st 4 component integer to be updated.</p>	Format:	U5	
Format:	U5			
2..n	127:0	<p>Entry</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">DX9_CONSTANTI_ENTRY</td> </tr> </table> <p>The four components of the nth float to be updated.</p>	Format:	DX9_CONSTANTI_ENTRY
Format:	DX9_CONSTANTI_ENTRY			



3DSTATE_DX9_CONSTANTI_VS

3DSTATE_DX9_CONSTANTI_VS			
Source:	RenderCS		
Length Bias:	2		
This command sets one or more DX9 constant integer registers for PS.			
Programming Notes			
<ul style="list-style-type: none"> The 3DSTATE_DX9_CONSTANTI_VS is a variable length command. Programming this command in batch buffer requires that all float, integer and boolean constants initialized prior to any commands or events that cause the constants to be written to memory. 			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	OpCode
	28:27	Command SubType	
		Default Value:	3h GFXPIPE_3D
	26:24	3D Command Opcode	
		Default Value:	0h GFXPIPE_PIPELINED
23:16	3D Command Sub Opcode		
	Default Value:	3Bh 3DSTATE_DX9_CONSTANTI_VS	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Format:	=n Total Length - 2	
	Value	Name	Description
	0h	[Default]	Excludes DWord (0,1)
0h-80h	multiples of 4		
1	31:16	Reserved	
		Format:	MBZ



3DSTATE_DX9_CONSTANTI_VS				
	15	<p>Global Constant Register</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> <p>When this bit is set the global constant register set will be updated. When this bit is clear the local constant register set will be updated.</p>	Format:	U1
	Format:	U1		
	14:5	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
4:0	<p>Constant Register Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U5</td> </tr> </table> <p>This field specifies the index of 1st 4 component integer to be updated.</p>	Format:	U5	
Format:	U5			
2..n	127:0	<p>Entry</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">DX9_CONSTANTI_ENTRY</td> </tr> </table> <p>The four components of the nth float to be updated.</p>	Format:	DX9_CONSTANTI_ENTRY
Format:	DX9_CONSTANTI_ENTRY			



3DSTATE_DX9_GENERATE_ACTIVE_PS

3DSTATE_DX9_GENERATE_ACTIVE_PS		
Source:	RenderCS	
Length Bias:	2	
<p>The 3DSTATE_DX9_GENERATE_ACTIVE_PS command is used to generate fixed functions' DX9 Constant Buffer. A DX9 Constant register is made active by writing it out to the constant buffer.</p>		
Programming Notes		
<p>Restriction: The global and local buffers are not initialized after reset. Any data written without being initialized will be undefined. DX9 constant buffers are written due to context save/restore or the Generate Active Command.</p>		
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: 0h 3DSTATE_PIPELINED Format: OpCode
	23:16	3D Command Sub Opcode
Default Value: 42h 3DSTATE_DX9_GENERATE_ACTIVE_PS Format: OpCode		
15:8	Reserved	
7:0	Format: MBZ	
	Dword Length	Default Value: 0h Excludes Dword (0,1) Format: =n
		Total Length - 2
1	31:24	Reserved
		Format: MBZ
1	23:13	Pointer to PS Constant Buffer
		Format: ConstantBufferOffset[23:13]BINDING_TABLE_STATE* Specifies the 8KB aligned address offset of the PS function's Dx9 constant buffer. This offset is relative to the DX9 Constant buffer Base Address.



3DSTATE_DX9_GENERATE_ACTIVE_PS											
12	DX9 Enable										
	Format:	Enable									
	Format:	U1									
	<p>When this bit is set, the Resource Streamer will generate the PS constant buffer according to the DX9 rules:</p> <ol style="list-style-type: none"> Valid local register are made active. Global register becomes active, unless the corresponding local register is valid. Local register valids are reset. <p>When this bit is cleared, the Resource Streamer will generate the PS constant buffer according to the DX8 rules:</p> <ol style="list-style-type: none"> Global register become active. Local register valids are reset. 										
Programming Notes											
In DX8 mode software will set all constants as globals, even ones locally defined within a shader.											
11	Clamp Enable										
	Format:	Enable									
	Format:	U1									
	<p>When this bit is set, the Resource Streamer will generate the PS constant buffer with the global values clamped to [-1,1]. When this bit is cleared, the Resource Streamer will generate the PS constant buffer without the global value clamped.</p>										
Programming Notes											
The clamping affects the values written out to the constant buffer and values read by the Gather engine for push constant buffer generation.											
10	Local Valid Bit Reset Enable										
	Format:	U1									
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>Disable resetting of the local valid bits on the generate active command</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Enable resetting of the local valid bits on the generate active command</td> </tr> </tbody> </table>		Value	Name	Description	0	Disable	Disable resetting of the local valid bits on the generate active command	1	Enable	Enable resetting of the local valid bits on the generate active command
	Value	Name	Description								
0	Disable	Disable resetting of the local valid bits on the generate active command									
1	Enable	Enable resetting of the local valid bits on the generate active command									
9	Buffer Write Disable										
	Format:	U1									
	<p>When this bit is set, the Resource Streamer will not write the values of the on-die registers to the constant buffer. The internal handshaking will take place as if it did.</p>										
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Enable Writes</td> <td>On-die register values are written out to the constant buffer.</td> </tr> <tr> <td>1</td> <td>Disable Writes</td> <td>On-die registers are not written out.</td> </tr> </tbody> </table>		Value	Name	Description	0	Enable Writes	On-die register values are written out to the constant buffer.	1	Disable Writes	On-die registers are not written out.
Value	Name	Description									
0	Enable Writes	On-die register values are written out to the constant buffer.									
1	Disable Writes	On-die registers are not written out.									



3DSTATE_DX9_GENERATE_ACTIVE_PS		
	8	Disable Produce to Command Stream Format: MBZ When set this bit will disable the produce to the command stream for the generation of a constant buffer. This bit must only be set by HW during a context save and restore. Otherwise it will cause the render command stream to hang waiting for a produce. During context save and restore command stream is not parsing the same command buffer.
	7:0	Reserved Format: MBZ



3DSTATE_DX9_GENERATE_ACTIVE_VS

3DSTATE_DX9_GENERATE_ACTIVE_VS		
Source:	RenderCS	
Length Bias:	2	
<p>The 3DSTATE_DX9_GENERATE_ACTIVE_VS command is used to generate fixed functions' DX9 Constant Buffer. A DX9 Constant register is made active by writing it out to the constant buffer.</p> <p>Programming Restriction: The global and local buffers are not initialized after reset. Any data written without being initialized will be undefined. DX9 constant buffers are written due to context save/restore or the Generate Active Command.</p>		
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: 0h 3DSTATE_PIPELINED Format: OpCode
	23:16	3D Command Sub Opcode
Default Value: 41h 3DSTATE_DX9_GENERATE_ACTIVE_VS Format: OpCode		
15:8	Reserved	
7:0	Format: MBZ	
	DWord Length	
Default Value: 0h Excludes DWord (0,1) Format: =n Total Length - 2		
1	31:24	Reserved
		Format: MBZ
23:13	Format: ConstantBufferOffset[23:13]	Pointer to VS Constant Buffer
		Specifies the 8KB aligned address offset of the VS function's Dx9 constant buffer. This offset is relative to the DX9 Constant buffer Base Address.



3DSTATE_DX9_GENERATE_ACTIVE_VS			
12	DX9 Enable		
	Format:	Enable	
	<p>When this bit is set, the Resource Streamer will generate the VS constant buffer according to the DX9 rules:</p> <ol style="list-style-type: none"> Valid local register are made active. Global register becomes active, unless the corresponding local register is valid. Local register valids are reset. <p>When this bit is cleared, the Resource Streamer will generate the VS constant buffer according to the DX8 rules:</p> <ol style="list-style-type: none"> Global register become active. Local register valids are reset. 		
	Programming Notes		
<p>In DX8 mode software will set all constants as globals, even ones locally defined within a shader.</p>			
11	Clamp Enable		
	Format:	Enable	
	<p>When this bit is set, the Resource Streamer will generate the VS constant buffer with the global values clamped to [-1,1]. When this bit is cleared, the Resource Streamer will generate the VS constant buffer without the global value clamped.</p>		
	Programming Notes		
<p>The clamping affects the values written out to the constant buffer and values read by the Gather engine for push constant buffer generation.</p>			
10	Local Valid Bit Reset Enable		
	Format:	Enable	
	Value	Name	Description
	0	Disable	Disable resetting of the local valid bits on the generate active command
	1	Enable	Enable resetting of the local valid bits on the generate active command
9	Buffer Write Disable		
	Format:	Disable	
	<p>When this bit is set, the Resource Streamer will not write the values of the on-die registers to the constant buffer. The internal handshaking will take place as if it did.</p>		
	Value	Name	Description
	0	Enable Writes	On-die register values are written out to the constant buffer.
1	Disable Writes	On-die registers are not written out.	



3DSTATE_DX9_GENERATE_ACTIVE_VS		
8	Disable Produce to Command Stream	
	Format:	MBZ
	When set this bit will disable the produce to the command stream for the generation of a constant buffer. This bit must only be set by HW during a context save and restore. Otherwise it will cause the render command stream to hang waiting for a produce. During context save and restore command stream is not parsing the same command buffer.	
7:0	Reserved	
	Format:	MBZ



3DSTATE_DX9_LOCAL_VALID_PS

3DSTATE_DX9_LOCAL_VALID_PS			
Source:		RenderCS	
Length Bias:		2	
This command sets the local valid bits for the DX9 Constant Buffer			
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:		0h GFXPIPE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	40h 3DSTATE_DX9_LOCAL_VALID_PS	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	Dword Length		
	Default Value:	9h Excludes Dword (0,1)	
	Format:	=n Total Length - 2	
1..8	31:0	Local ConstantF Valid Bits	
		Format:	U32
Each bit field when set indicates that the corresponding local register is valid. When the bit is clear it indicates the local register is invalid.			
9	31:0	Local ConstantI Valid Bits	
		Format:	U32
Each bit field when set indicates that the corresponding local register is valid. When the bit is clear it indicates the local register is invalid.			
10	31:16	Reserved	
		Format:	MBZ
	15:0	Local ConstantB Valid Bits	
Format:		U16	
Each bit field when set indicates that the corresponding local register is valid. When the bit is clear it indicates the local register is invalid.			



3DSTATE_DX9_LOCAL_VALID_VS

3DSTATE_DX9_LOCAL_VALID_VS			
Source:	RenderCS		
Length Bias:	2		
This command sets the local valid bits for the DX9 Constant Buffer			
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:		0h GFXPIPE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	3Fh 3DSTATE_DX9_LOCAL_VALID_VS	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	9h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1..8	31:0	Local ConstantF Valid Bits	
		Format:	U32
Each bit field when set indicates that the corresponding local register is valid. When the bit is clear it indicates the local register is invalid.			
9	31:0	Local ConstantI Valid Bits	
		Format:	U32
Each bit field when set indicates that the corresponding local register is valid. When the bit is clear it indicates the local register is invalid.			
10	31:16	Reserved	
		Format:	MBZ
	15:0	Local ConstantB Valid Bits	
Format:		U16	
Each bit field when set indicates that the corresponding local register is valid. When the bit is clear it indicates the local register is invalid.			



3DSTATE_GATHER_CONSTANT_DS

3DSTATE_GATHER_CONSTANT_DS		
Source:	RenderCS	
Length Bias:	2	
<p>This command uses the constant buffer binding table entries to reference constant buffer surface states for the DS unit. The constant data in these is gathered and packed according to a gather table contained in this command.</p>		
Programming Notes		
<p>The HW generated binding table must be enabled to use this command.</p>		
<p>The length of the gather table is derived from the total length of the command. The command length is in DWords, but the gather table entries are 16 bits in length. If there is an unused odd entry at the end of the command the channel mask should be set to all 0s.</p>		
<p>When a 3DSTATE_GATHER_CONSTANT_* command is used there must be a matching 3DSTATE_CONSTANT_*. Furthermore the 3DSTATE_CONSTANT_* must occur in the same order as the 3DSTATE_GATHER_CONSTANT_*. For example if a 3DSTATE_GATHER_CONSTANT_VS occurs before a 3DSTATE_GATHER_CONSTANT_PS, then the 3DSTATE_CONSTANT_VS must occur before the 3DSTATE_CONSTANT_PS.</p>		
<p>If Gather pool is enabled, there must be a corresponding 3DSTATE_GATHER_CONSTANT command with any 3DSTATE_CONSTANT for any particular shader. To avoid any update to the Gather pool, and yet program the 3DSTATE_CONSTANT for a particular shader, send a 3DSTATE_GATHER_CONSTANT command with all valid bits set to zero.</p>		
<p>The 3DSTATE_GATHER_CONSTANT_* command is not committed to the resource streamer engine until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command. For example, the 3DSTATE_GATHER_CONSTANT_VS command will not actually generate a buffer in memory until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the resource streamer.</p>		
<p>The following commands must be executed after any 3DSTATE_GATHER_CONSTANT_* command that has Constant Buffer Valid greater than zero: •(N times, minimum number is 4) MI_RS_STORE_DATA_IMM –To force engine idle before executing the next instruction. Write must occur to address that will not corrupt memory: •Resource Streamer Flush = 1 •3DSTATE_GATHER_CONSTANT_* (Ensures correct timing of sync between resource streamer and render pipeline) •The Constant Buffer Valid field should be zero and the Dword length equal to 1h. •3DSTATE_CONSTANT_*: •All values match the previous 3DSTATE_CONSTANT_*</p>		
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



3DSTATE_GATHER_CONSTANT_DS											
	26:24	3D Command Opcode Default Value: 0h 3DSTATE_PIPELINED Format: OpCode									
	23:16	3D Command Sub Opcode Default Value: 37h 3DSTATE_GATHER_CONSTANT_DS Format: OpCode									
	15:8	Reserved Format: MBZ									
	7:0	DWord Length Format: =n Total Length - 2 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 55%;">Name</th> <th style="width: 30%;">Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>DWORD_COUNT_n [Default]</td> <td>excludes DWords 0,1</td> </tr> <tr> <td>[1,128]</td> <td>Range</td> <td>1-128 Entries</td> </tr> </tbody> </table>	Value	Name	Description	1	DWORD_COUNT_n [Default]	excludes DWords 0,1	[1,128]	Range	1-128 Entries
Value	Name	Description									
1	DWORD_COUNT_n [Default]	excludes DWords 0,1									
[1,128]	Range	1-128 Entries									
1	31:16	Constant Buffer Valid Format: U16 This field specifies which of the 16 constant buffers are used in the push constant gather. If a bit is set it indicates the corresponding constant buffer is used. If a bit is clear it indicates the corresponding constant buffer is not used. If this field is zero it indicate that the gather buffer is not used.									
	15:12	Constant Buffer Binding Table Block Format: U4 This field specifies the 16 entry block constant buffer in the binding table. The constant buffer entry block must be aligned on a 16 entry boundary.									
	11:2	Reserved Format: MBZ									
	1	Update Gather Table Only Format: U1 When this bit is set, the Resource Streamer will update the on chip buffer of Gather Index values. Any following Binding Table Pointer will not commit this command to the Gather Engine. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 55%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Commit Gather</td> <td>Update Gather Table and Commit Gather on next BTP.</td> </tr> <tr> <td>1</td> <td>Non-Commit Gather</td> <td>Update Gather Table Only.</td> </tr> </tbody> </table>	Value	Name	Description	0	Commit Gather	Update Gather Table and Commit Gather on next BTP.	1	Non-Commit Gather	Update Gather Table Only.
	Value	Name	Description								
0	Commit Gather	Update Gather Table and Commit Gather on next BTP.									
1	Non-Commit Gather	Update Gather Table Only.									
0	Reserved Format: MBZ										
2	31:23	Reserved Format: MBZ									



3DSTATE_GATHER_CONSTANT_DS												
	22:6	Gather Buffer Offset Format: <table border="1" style="display: inline-table;"><tr><td>GatherBufferOffset[22:6]</td></tr></table> This field specifies the offset of the gather buffer within the Gather Pool <table border="1" style="width: 100%; text-align: center;"><tr><td>Programming Notes</td></tr></table> SW increments the offset by the size of the gather buffer in 512 bit units for each gather buffer generated.	GatherBufferOffset[22:6]	Programming Notes								
	GatherBufferOffset[22:6]											
	Programming Notes											
	5	Constant Buffer Dx9 Generate Stall Format: <table border="1" style="display: inline-table;"><tr><td>Enable</td></tr></table> When this bit is set the resource streamer will wait for the Dx9 constant buffer generator to be done before issuing this command to ensure buffer synchronization.	Enable									
	Enable											
4	Reserved Format: <table border="1" style="display: inline-table;"><tr><td>MBZ</td></tr></table>	MBZ										
MBZ												
3	On-Die Table Format: <table border="1" style="display: inline-table;"><tr><td>U1</td></tr></table> This bit controls whether the on-die table is loaded or read. <table border="1" style="width: 100%;"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>0h</td><td>Load</td><td>Load the on-die table w/ table contained in this command and do the gather using the table in this command.</td></tr><tr><td>1h</td><td>Read</td><td>Read the on-die table and use its contents for doing the gather command.</td></tr></tbody></table> <table border="1" style="width: 100%; text-align: center;"><tr><td>Programming Notes</td></tr></table> The HW will overwrite DW1 and DW2 of the previous GATHER with the current DW1 and DW2. If using the same gather tables, then the valid bits must be still in sync with the indexes used in the previous GATHER table. The Dword Length field must be set to 1h when this field is enabled.	U1	Value	Name	Description	0h	Load	Load the on-die table w/ table contained in this command and do the gather using the table in this command.	1h	Read	Read the on-die table and use its contents for doing the gather command.	Programming Notes
U1												
Value	Name	Description										
0h	Load	Load the on-die table w/ table contained in this command and do the gather using the table in this command.										
1h	Read	Read the on-die table and use its contents for doing the gather command.										
Programming Notes												
2:0	Reserved Format: <table border="1" style="display: inline-table;"><tr><td>MBZ</td></tr></table>	MBZ										
MBZ												
3..n	31:16	Entry_1 Format: <table border="1" style="display: inline-table;"><tr><td>GATHER_CONSTANT_ENTRY</td></tr></table>	GATHER_CONSTANT_ENTRY									
	GATHER_CONSTANT_ENTRY											
15:0	Entry_0 Format: <table border="1" style="display: inline-table;"><tr><td>GATHER_CONSTANT_ENTRY</td></tr></table>	GATHER_CONSTANT_ENTRY										
GATHER_CONSTANT_ENTRY												



3DSTATE_GATHER_CONSTANT_GS

3DSTATE_GATHER_CONSTANT_GS			
Source:	RenderCS		
Length Bias:	2		
<p>This command uses the constant buffer binding table entries to reference constant buffer surface states for GS unit. The constant data in these is gathered and packed according to a gather table contained in this command.</p>			
Programming Notes			
<p>The HW generated binding table must be enabled to use this command.</p>			
<p>The length of the gather table is derived from the total length of the command. The command length is in DWords, but the gather table entries are 16 bits in length. If there is an unused odd entry at the end of the command the channel mask should be set to all 0s.</p>			
<p>When a 3DSTATE_GATHER_CONSTANT_* command is used there must be a matching 3DSTATE_CONSTANT_*. Furthermore the 3DSTATE_CONSTANT_* must occur in the same order as the 3DSTATE_GATHER_CONSTANT_*. For example if a 3DSTATE_GATHER_CONSTANT_VS occurs before a 3DSTATE_GATHER_CONSTANT_PS, then the 3DSTATE_CONSTANT_VS must occur before the 3DSTATE_CONSTANT_PS.</p>			
<p>If Gather pool is enabled, there must be a corresponding 3DSTATE_GATHER_CONSTANT command with any 3DSTATE_CONSTANT for any particular shader. To avoid any update to the Gather pool, and yet program the 3DSTATE_CONSTANT for a particular shader, send a 3DSTATE_GATHER_CONSTANT command with all valid bits set to zero.</p>			
<p>The 3DSTATE_GATHER_CONSTANT_* command is not committed to the resource streamer engine until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command. For example, the 3DSTATE_GATHER_CONSTANT_VS command will not actually generate a buffer in memory until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the resource streamer.</p>			
<p>The following commands must be executed after any 3DSTATE_GATHER_CONSTANT_* command that has Constant Buffer Valid greater than zero:</p> <ul style="list-style-type: none"> •(N times, minimum number is 4) MI_RS_STORE_DATA_IMM –To force engine idle before executing the next instruction. Write must occur to address that will not corrupt memory: •Resource Streamer Flush = 1 •3DSTATE_GATHER_CONSTANT_* (Ensures correct timing of sync between resource streamer and render pipeline) •The Constant Buffer Valid field should be zero and the Dword length equal to 1h. •3DSTATE_CONSTANT_*: •All values match the previous 3DSTATE_CONSTANT_* 			
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



3DSTATE_GATHER_CONSTANT_GS													
1	26:24	3D Command Opcode <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>0h 3DSTATE_PIPELINED</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	0h 3DSTATE_PIPELINED	Format:	OpCode							
	Default Value:	0h 3DSTATE_PIPELINED											
	Format:	OpCode											
	23:16	3D Command Sub Opcode <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>35h 3DSTATE_GATHER_CONSTANT_GS</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	35h 3DSTATE_GATHER_CONSTANT_GS	Format:	OpCode							
	Default Value:	35h 3DSTATE_GATHER_CONSTANT_GS											
	Format:	OpCode											
	15:8	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ											
	7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>=n</td> </tr> </table> <p>Total Length - 2</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 45%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>DWORD_COUNT_n [Default]</td> <td>excludes DWords 0,1</td> </tr> <tr> <td>[1,128]</td> <td>Range</td> <td>1-128 Entries</td> </tr> </tbody> </table>	Format:	=n	Value	Name	Description	1	DWORD_COUNT_n [Default]	excludes DWords 0,1	[1,128]	Range	1-128 Entries
	Format:	=n											
	Value	Name	Description										
	1	DWORD_COUNT_n [Default]	excludes DWords 0,1										
[1,128]	Range	1-128 Entries											
31:16	Constant Buffer Valid <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>U16</td> </tr> </table> <p>This field specifies which of the 16 constant buffers are used in the push constant gather. If a bit is set it indicates the corresponding constant buffer is used. If a bit is clear it indicates the corresponding constant buffer is not used. If this field is zero it indicate that the gather buffer is not used.</p>	Format:	U16										
Format:	U16												
15:12	Constant Buffer Binding Table Block <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the 16 entry block constant buffer in the binding table. The constant buffer entry block must be aligned on a 16 entry boundary.</p>	Format:	U4										
Format:	U4												
11:2	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
1	Update Gather Table Only <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>U1</td> </tr> </table> <p>When this bit is set, the Resource Streamer will update the on chip buffer of Gather Index values. Any following Binding Table Pointer will not commit this command to the Gather Engine.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 55%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Commit Gather</td> <td>Update Gather Table and Commit Gather on next BTP.</td> </tr> <tr> <td>1</td> <td>Non-Commit Gather</td> <td>Update Gather Table Only.</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	0	Commit Gather	Update Gather Table and Commit Gather on next BTP.	1	Non-Commit Gather	Update Gather Table Only.	
Format:	U1												
Value	Name	Description											
0	Commit Gather	Update Gather Table and Commit Gather on next BTP.											
1	Non-Commit Gather	Update Gather Table Only.											
0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												



3DSTATE_GATHER_CONSTANT_GS											
2	31:23	Reserved Format: _____ MBZ									
	22:6	Gather Buffer Offset Format: _____ GatherBufferOffset[22:6] This field specifies the offset of the gather buffer within the Gather Pool. <div style="text-align: center; background-color: #e1eef6; padding: 2px;">Programming Notes</div> SW increments the offset by the size of the gather buffer in 512 bit units for each gather buffer generated.									
	5	Constant Buffer Dx9 Generate Stall Format: _____ Enable When this bit is set the resource streamer will wait for the Dx9 constant buffer generator to be done before issuing this command to ensure buffer synchronization.									
	4	Reserved Format: _____ MBZ									
	3	On-Die Table Format: _____ U1 This bit controls whether the on-die table is loaded or read. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="text-align: left;">Value</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Load</td> <td>Load the on-die table w/ table contained in this command and do the gather using the table in this command.</td> </tr> <tr> <td>1h</td> <td>Read</td> <td>Read the on-die table and use its contents for doing the gather command.</td> </tr> </tbody> </table> <div style="text-align: center; background-color: #e1eef6; padding: 2px;">Programming Notes</div> The HW will overwrite DW1 and DW2 of the previous GATHER with the current DW1 and DW2. If using the same gather tables, then the valid bits must be still in sync with the indexes used in the previous GATHER table. The Dword Length field must be set to 1h when this field is enabled.	Value	Name	Description	0h	Load	Load the on-die table w/ table contained in this command and do the gather using the table in this command.	1h	Read	Read the on-die table and use its contents for doing the gather command.
	Value	Name	Description								
0h	Load	Load the on-die table w/ table contained in this command and do the gather using the table in this command.									
1h	Read	Read the on-die table and use its contents for doing the gather command.									
2:0	Reserved Format: _____ MBZ										
3..n	31:16	Entry_1 Format: _____ GATHER_CONSTANT_ENTRY									
	15:0	Entry_0 Format: _____ GATHER_CONSTANT_ENTRY									



3DSTATE_GATHER_CONSTANT_HS

3DSTATE_GATHER_CONSTANT_HS		
Source:	RenderCS	
Length Bias:	2	
<p>This command uses the constant buffer binding table entries to reference constant buffer surface states for HS unit. The constant data in these is gathered and packed according to a gather table contained in this command.</p>		
Programming Notes		
<p>The HW generated binding table must be enabled to use this command.</p>		
<p>The length of the gather table is derived from the total length of the command. The command length is in DWords, but the gather table entries are 16 bits in length. If there is an unused odd entry at the end of the command the channel mask should be set to all 0s.</p>		
<p>When a 3DSTATE_GATHER_CONSTANT_* command is used there must be a matching 3DSTATE_CONSTANT_*. Furthermore the 3DSTATE_CONSTANT_* must occur in the same order as the 3DSTATE_GATHER_CONSTANT_*. For example if a 3DSTATE_GATHER_CONSTANT_VS occurs before a 3DSTATE_GATHER_CONSTANT_PS, then the 3DSTATE_CONSTANT_VS must occur before the 3DSTATE_CONSTANT_PS.</p>		
<p>If Gather pool is enabled, there must be a corresponding 3DSTATE_GATHER_CONSTANT command with any 3DSTATE_CONSTANT for any particular shader. To avoid any update to the Gather pool, and yet program the 3DSTATE_CONSTANT for a particular shader, send a 3DSTATE_GATHER_CONSTANT command with all valid bits set to zero.</p>		
<p>The 3DSTATE_GATHER_CONSTANT_* command is not committed to the resource streamer engine until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command. For example, the 3DSTATE_GATHER_CONSTANT_VS command will not actually generate a buffer in memory until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the resource streamer.</p>		
<p>The following commands must be executed after any 3DSTATE_GATHER_CONSTANT_* command that has Constant Buffer Valid greater than zero:</p> <ul style="list-style-type: none"> •(N times, minimum number is 4) MI_RS_STORE_DATA_IMM –To force engine idle before executing the next instruction. Write must occur to address that will not corrupt memory: •Resource Streamer Flush = 1 •3DSTATE_GATHER_CONSTANT_* (Ensures correct timing of sync between resource streamer and render pipeline) •The Constant Buffer Valid field should be zero and the Dword length equal to 1h. •3DSTATE_CONSTANT_*: •All values match the previous 3DSTATE_CONSTANT_* 		
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



3DSTATE_GATHER_CONSTANT_HS											
	26:24	3D Command Opcode Default Value: 0h 3DSTATE_PIPELINED Format: OpCode									
	23:16	3D Command Sub Opcode Default Value: 36h 3DSTATE_GATHER_CONSTANT_HS Format: OpCode									
	15:8	Reserved Format: MBZ									
	7:0	DWord Length Format: =n Total Length - 2 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 55%;">Name</th> <th style="width: 30%;">Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>DWORD_COUNT_n [Default]</td> <td>excludes DWords 0,1</td> </tr> <tr> <td>[1,128]</td> <td>Range</td> <td>1-128 Entries</td> </tr> </tbody> </table>	Value	Name	Description	1	DWORD_COUNT_n [Default]	excludes DWords 0,1	[1,128]	Range	1-128 Entries
Value	Name	Description									
1	DWORD_COUNT_n [Default]	excludes DWords 0,1									
[1,128]	Range	1-128 Entries									
1	31:16	Constant Buffer Valid Format: U16 This field specifies which of the 16 constant buffers are used in the push constant gather. If a bit is set it indicates the corresponding constant buffer is used. If a bit is clear it indicates the corresponding constant buffer is not used. If this field is zero it indicate that the gather buffer is not used.									
	15:12	Constant Buffer Binding Table Block Format: U4 This field specifies the 16 entry block constant buffer in the binding table. The constant buffer entry block must be aligned on a 16 entry boundary.									
	11:2	Reserved Format: MBZ									
	1	Update Gather Table Only Format: U1 When this bit is set, the Resource Streamer will update the on chip buffer of Gather Index values. Any following Binding Table Pointer will not commit this command to the Gather Engine. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 55%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Commit Gather</td> <td>Update Gather Table and Commit Gather on next BTP.</td> </tr> <tr> <td>1</td> <td>Non-Commit Gather</td> <td>Update Gather Table Only.</td> </tr> </tbody> </table>	Value	Name	Description	0	Commit Gather	Update Gather Table and Commit Gather on next BTP.	1	Non-Commit Gather	Update Gather Table Only.
	Value	Name	Description								
0	Commit Gather	Update Gather Table and Commit Gather on next BTP.									
1	Non-Commit Gather	Update Gather Table Only.									
0	Reserved Format: MBZ										
2	31:23	Reserved Format: MBZ									



3DSTATE_GATHER_CONSTANT_HS											
	22:6	Gather Buffer Offset Format: <table border="1" style="display: inline-table;"><tr><td>GatherBufferOffset[22:6]</td></tr></table> This field specifies the offset of the gather buffer within the Gather Pool. <div style="text-align: center;">Programming Notes</div> SW increments the offset by the size of the gather buffer in 512 bit units for each gather buffer generated.	GatherBufferOffset[22:6]								
	GatherBufferOffset[22:6]										
	5	Constant Buffer Dx9 Generate Stall Format: <table border="1" style="display: inline-table;"><tr><td>Enable</td></tr></table> When this bit is set the resource streamer will wait for the Dx9 constant buffer generator to be done before issuing this command to ensure buffer synchronization.	Enable								
	Enable										
	4	Reserved Format: <table border="1" style="display: inline-table;"><tr><td>MBZ</td></tr></table>	MBZ								
MBZ											
3	On-Die Table Format: <table border="1" style="display: inline-table;"><tr><td>U1</td></tr></table> This bit controls whether the on-die table is loaded or read. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Load</td> <td>Load the on-die table w/ table contained in this command and do the gather using the table in this command.</td> </tr> <tr> <td>1h</td> <td>Read</td> <td>Read the on-die table and use its contents for doing the gather command.</td> </tr> </tbody> </table> <div style="text-align: center;">Programming Notes</div> The HW will overwrite DW1 and DW2 of the previous GATHER with the current DW1 and DW2. If using the same gather tables, then the valid bits must be still in sync with the indexes used in the previous GATHER table. The Dword Length field must be set to 1h when this field is enabled.	U1	Value	Name	Description	0h	Load	Load the on-die table w/ table contained in this command and do the gather using the table in this command.	1h	Read	Read the on-die table and use its contents for doing the gather command.
U1											
Value	Name	Description									
0h	Load	Load the on-die table w/ table contained in this command and do the gather using the table in this command.									
1h	Read	Read the on-die table and use its contents for doing the gather command.									
2:0	Reserved Format: <table border="1" style="display: inline-table;"><tr><td>MBZ</td></tr></table>	MBZ									
MBZ											
3..n	31:16	Entry_1 Format: <table border="1" style="display: inline-table;"><tr><td>GATHER_CONSTANT_ENTRY</td></tr></table>	GATHER_CONSTANT_ENTRY								
	GATHER_CONSTANT_ENTRY										
15:0	Entry_0 Format: <table border="1" style="display: inline-table;"><tr><td>GATHER_CONSTANT_ENTRY</td></tr></table>	GATHER_CONSTANT_ENTRY									
GATHER_CONSTANT_ENTRY											



3DSTATE_GATHER_CONSTANT_PS

3DSTATE_GATHER_CONSTANT_PS			
Source:	RenderCS		
Length Bias:	2		
<p>This command uses the constant buffer binding table entries to reference constant buffer surface states for PS unit. The constant data in these is gathered and packed according to a gather table contained in this command.</p>			
Programming Notes			
<p>The HW generated binding table must be enabled to use this command.</p>			
<p>The length of the gather table is derived from the total length of the command. The command length is in DWords, but the gather table entries are 16 bits in length. If there is an unused odd entry at the end of the command the channel mask should be set to all 0s.</p>			
<p>When a 3DSTATE_GATHER_CONSTANT_* command is used there must be a matching 3DSTATE_CONSTANT_*. Furthermore the 3DSTATE_CONSTANT_* must occur in the same order as the 3DSTATE_GATHER_CONSTANT_*. For example if a 3DSTATE_GATHER_CONSTANT_VS occurs before a 3DSTATE_GATHER_CONSTANT_PS, then the 3DSTATE_CONSTANT_VS must occur before the 3DSTATE_CONSTANT_PS.</p>			
<p>If Gather pool is enabled, there must be a corresponding 3DSTATE_GATHER_CONSTANT command with any 3DSTATE_CONSTANT for any particular shader. To avoid any update to the Gather pool, and yet program the 3DSTATE_CONSTANT for a particular shader, send a 3DSTATE_GATHER_CONSTANT command with all valid bits set to zero.</p>			
<p>The 3DSTATE_GATHER_CONSTANT_* command is not committed to the resource streamer engine until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command. For example, the 3DSTATE_GATHER_CONSTANT_VS command will not actually generate a buffer in memory until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the resource streamer.</p>			
<p>The following commands must be executed after any 3DSTATE_GATHER_CONSTANT_* command that has Constant Buffer Valid greater than zero: •(N times, minimum number is 4) MI_RS_STORE_DATA_IMM –To force engine idle before executing the next instruction. Write must occur to address that will not corrupt memory: •Resource Streamer Flush = 1 •3DSTATE_GATHER_CONSTANT_* (Ensures correct timing of sync between resource streamer and render pipeline) •The Constant Buffer Valid field should be zero and the Dword length equal to 1h. •3DSTATE_CONSTANT_*: •All values match the previous 3DSTATE_CONSTANT_*</p>			
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:	0h 3DSTATE_PIPELINED



3DSTATE_GATHER_CONSTANT_PS		
		Format: OpCode
23:16	3D Command Sub Opcode	
	Default Value:	38h 3DSTATE_GATHER_CONSTANT_PS
	Format:	OpCode
15:8	Reserved	
	Format:	MBZ
7:0	DWord Length	
	Format:	=n
	Total Length - 2	
	Value	Name Description
	1	DWORD_COUNT_n [Default] excludes DWords 0,1
	[1,128]	Range 1-128 Entries
1	31:16	Constant Buffer Valid
	Format:	U16
	This field specifies which of the 16 constant buffers are used in the push constant gather. If a bit is set it indicates the corresponding constant buffer is used. If a bit is clear it indicates the corresponding constant buffer is not used. If this field is zero it indicate that the gather buffer is not used.	
	15:12	Constant Buffer Binding Table Block
	Format:	U4
	This field specifies the 16 entry block constant buffer in the binding table. The constant buffer entry block must be aligned on a 16 entry boundary.	
	11:2	Reserved
	Format:	MBZ
	1	Update Gather Table Only
	Format:	U1
	When this bit is set, the Resource Streamer will update the on chip buffer of Gather Index values. Any following Binding Table Pointer will not commit this command to the Gather Engine.	
	Value	Name Description
	0	Commit Gather Update Gather Table and Commit Gather on next BTP.
	1	Non-Commit Gather Update Gather Table Only.
	0	DX9 On-Die Register Read Enable
	Format:	Enable
	When this bit is set, the Resource Streamer will not read out to the constant buffer, but instead it will read from the on-die registers. The internal handshaking will take place as if it did perform the read from the constant buffer. The Constant Buffer Offset is the offset into the on-die register and the Binding Table Index Offset will cause it to be a 4KB offset if the value is '1'. If the Binding Table Index Offset is '0' then the engine will read from the lower 4KB of the	



3DSTATE_GATHER_CONSTANT_PS														
		constant buffer. This bit only has effect if the Dx9 enable is set.												
2	31:23	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
	Format:	MBZ												
22:6	Gather Buffer Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GatherBufferOffset[22:6]</td> </tr> </table> <p>This field specifies the offset of the gather buffer within the Gather Pool.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </table> <p>SW increments the offset by the size of the gather buffer in 512 bit units for each gather buffer generated.</p>	Format:	GatherBufferOffset[22:6]	Programming Notes										
Format:	GatherBufferOffset[22:6]													
Programming Notes														
5		Constant Buffer Dx9 Generate Stall <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>When this bit is set the resource streamer will wait for the Dx9 constant buffer generator to be done before issuing this command to ensure buffer synchronization.</p>	Format:	Enable										
	Format:	Enable												
	Constant Buffer Dx9 Enable <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>When this bit is set it indicates that the constant buffer is a HW generated Dx9 constant buffer. The resource streamer will wait for the Dx9 constant buffer generator to be done before issuing this command to ensure buffer synchronization. Additionally the Dx9 constant buffers are a single buffer but larger than 4KB. Internally the HW will treat the DX9 buffer as 2 constant buffers. When this bit is enable only the 1st constant buffer valid bit is set. The 2nd constant buffer surface pointer will automatically be the 1st pointer + 4KB.</p>	Format:	Enable											
Format:	Enable													
3		On-Die Table <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U1</td> </tr> </table> <p>This bit controls whether the on-die table is loaded or read.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Load</td> <td>Load the on-die table w/ table contained in this command and do the gather using the table in this command.</td> </tr> <tr> <td>1h</td> <td>Read</td> <td>Read the on-die table and use its contents for doing the gather command.</td> </tr> </tbody> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </table> <p>The HW will overwrite DW1 and DW2 of the previous GATHER with the current DW1 and DW2. If using the same gather tables, then the valid bits must be still in sync with the indexes used in the previous GATHER table.</p> <p>The Dword Length field must be set to 1h when this field is enabled.</p>	Format:	U1	Value	Name	Description	0h	Load	Load the on-die table w/ table contained in this command and do the gather using the table in this command.	1h	Read	Read the on-die table and use its contents for doing the gather command.	Programming Notes
	Format:	U1												
	Value	Name	Description											
	0h	Load	Load the on-die table w/ table contained in this command and do the gather using the table in this command.											
	1h	Read	Read the on-die table and use its contents for doing the gather command.											
Programming Notes														
2:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ											
Format:	MBZ													
3..n	31:16	Entry_1												



3DSTATE_GATHER_CONSTANT_PS			
		Format:	GATHER_CONSTANT_ENTRY
	15:0	Entry_0	
		Format:	GATHER_CONSTANT_ENTRY



3DSTATE_GATHER_CONSTANT_VS

3DSTATE_GATHER_CONSTANT_VS		
Source:	RenderCS	
Length Bias:	2	
<p>This command uses the constant buffer binding table entries to reference constant buffer surface states for VS unit. The constant data in these is gathered and packed according to a gather table contained in this command.</p>		
Programming Notes		
<p>The HW generated binding table must be enabled to use this command.</p>		
<p>The length of the gather table is derived from the total length of the command. The command length is in DWords, but the gather table entries are 16 bits in length. If there is an unused odd entry at the end of the command the channel mask should be set to all 0s.</p>		
<p>When a 3DSTATE_GATHER_CONSTANT_* command is used there must be a matching 3DSTATE_CONSTANT_*. Furthermore the 3DSTATE_CONSTANT_* must occur in the same order as the 3DSTATE_GATHER_CONSTANT_*. For example if a 3DSTATE_GATHER_CONSTANT_VS occurs before a 3DSTATE_GATHER_CONSTANT_PS, then the 3DSTATE_CONSTANT_VS must occur before the 3DSTATE_CONSTANT_PS.</p>		
<p>If Gather pool is enabled, there must be a corresponding 3DSTATE_GATHER_CONSTANT command with any 3DSTATE_CONSTANT for any particular shader. To avoid any update to the Gather pool, and yet program the 3DSTATE_CONSTANT for a particular shader, send a 3DSTATE_GATHER_CONSTANT command with all valid bits set to zero.</p>		
<p>The 3DSTATE_GATHER_CONSTANT_* command is not committed to the resource streamer engine until the corresponding (same shader) 3DSTATE_BINDING_TABLE_POINTER_* command. For example, the 3DSTATE_GATHER_CONSTANT_VS command will not actually generate a buffer in memory until the 3DSTATE_BINDING_TABLE_POINTERS_VS is parsed by the resource streamer.</p>		
Workaround		
<p>The following commands must be executed after any 3DSTATE_GATHER_CONSTANT_* command that has Constant Buffer Valid greater than zero: •(N times, minimum number is 4) MI_RS_STORE_DATA_IMM –To force engine idle before executing the next instruction. Write must occur to address that will not corrupt memory: •Resource Streamer Flush = 1 •3DSTATE_GATHER_CONSTANT_* (Ensures correct timing of sync between resource streamer and render pipeline) •The Constant Buffer Valid field should be zero and the Dword length equal to 1h. •3DSTATE_CONSTANT_*: •All values match the previous 3DSTATE_CONSTANT_*</p>		
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



3DSTATE_GATHER_CONSTANT_VS												
	26:24	3D Command Opcode <table border="1"> <tr> <td>Default Value:</td> <td>0h 3DSTATE_PIPELINED</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	0h 3DSTATE_PIPELINED	Format:	OpCode						
	Default Value:	0h 3DSTATE_PIPELINED										
	Format:	OpCode										
	23:16	3D Command Sub Opcode <table border="1"> <tr> <td>Default Value:</td> <td>34h 3DSTATE_GATHER_CONSTANT_VS</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	34h 3DSTATE_GATHER_CONSTANT_VS	Format:	OpCode						
Default Value:	34h 3DSTATE_GATHER_CONSTANT_VS											
Format:	OpCode											
15:8	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
7:0	DWord Length <table border="1"> <tr> <td>Format:</td> <td>=n</td> </tr> </table> <p>Total Length - 2</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>DWORD_COUNT_n [Default]</td> <td>excludes DWords 0,1</td> </tr> <tr> <td>[1,128]</td> <td>Range</td> <td>1-128 Entries</td> </tr> </tbody> </table>	Format:	=n	Value	Name	Description	0	DWORD_COUNT_n [Default]	excludes DWords 0,1	[1,128]	Range	1-128 Entries
Format:	=n											
Value	Name	Description										
0	DWORD_COUNT_n [Default]	excludes DWords 0,1										
[1,128]	Range	1-128 Entries										
1	31:16	Constant Buffer Valid <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>This field specifies which of the 16 constant buffers are used in the push constant gather. If a bit is set it indicates the corresponding constant buffer is used. If a bit is clear it indicates the corresponding constant buffer is not used. If this field is zero it indicate that the gather buffer is not used.</p>	Format:	U16								
	Format:	U16										
	15:12	Constant Buffer Binding Table Block <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the 16 entry block constant buffer in the binding table. The constant buffer entry block must be aligned on a 16 entry boundary.</p>	Format:	U4								
	Format:	U4										
11:2	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
1	Update Gather Table Only <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>When this bit is set, the Resource Streamer will update the on chip buffer of Gather Index values. Any following Binding Table Pointer will not commit this command to the Gather Engine.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Commit Gather</td> <td>Update Gather Table and Commit Gather on next BTP.</td> </tr> <tr> <td>1</td> <td>Non-Commit Gather</td> <td>Update Gather Table Only.</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	0	Commit Gather	Update Gather Table and Commit Gather on next BTP.	1	Non-Commit Gather	Update Gather Table Only.
Format:	U1											
Value	Name	Description										
0	Commit Gather	Update Gather Table and Commit Gather on next BTP.										
1	Non-Commit Gather	Update Gather Table Only.										



3DSTATE_GATHER_CONSTANT_VS			
0	DX9 On-Die Register Read Enable		
	Format:	Enable	
<p>When this bit is set, the Resource Streamer will not read out to the constant buffer, but instead it will read from the on-die registers. The internal handshaking will take place as if it did perform the read from the constant buffer. The Constant Buffer Offset is the offset into the on-die register and the Binding Table Index Offset will cause it to be a 4KB offset if the value is '1'. If the Binding Table Index Offset is '0' then the engine will read from the lower 4KB of the constant buffer. This bit only has effect if the Dx9 enable is set.</p>			
2	31:23	Reserved	
	Format:	MBZ	
	22:6	Gather Buffer Offset	
	Format:	GatherBufferOffset[22:6]	
	<p>This field specifies the offset of the gather buffer within the Gather Pool.</p> <p style="text-align: center;">Programming Notes</p> <p>SW increments the offset by the size of the gather buffer in 512 bit units for each gather buffer generated.</p>		
5	Constant Buffer Dx9 Generate Stall		
	Format:	Enable	
<p>When this bit is set the resource streamer will wait for the Dx9 constant buffer generator to be done before issuing this command to ensure buffer synchronization.</p>			
4	Constant Buffer Dx9 Enable		
	Format:	Enable	
<p>When this bit is set it indicates that the constant buffer is a HW generated Dx9 constant buffer. The resource streamer will wait for the Dx9 constant buffer generator to be done before issuing this command to ensure buffer synchronization. Additionally the Dx9 constant buffers are a single buffer but larger than 4KB. Internally the HW will treat the DX9 buffer as 2 constant buffers. When this bit is enable only the 1st constant buffer valid bit is set. The 2nd constant buffer surface pointer will automatically be the 1st pointer + 4KB.</p>			
3	On-Die Table		
	Format:	U1	
	<p>This bit controls whether the on-die table is loaded or read.</p>		
	Value	Name	Description
	0h	Load	Load the on-die table w/ table contained in this command and do the gather using the table in this command.
	1h	Read	Read the on-die table and use its contents for doing the gather command.
Programming Notes			
<p>The HW will overwrite DW1 and DW2 of the previous GATHER with the current DW1 and DW2. If using the same gather tables, then the valid bits must be still in sync with the indexes used in</p>			



3DSTATE_GATHER_CONSTANT_VS		
		the previous GATHER table. The Dword Length field must be set to 1h when this field is enabled.
	2:0	Reserved Format: MBZ
3..n	31:16	Entry_1 Format: GATHER_CONSTANT_ENTRY
	15:0	Entry_0 Format: GATHER_CONSTANT_ENTRY



3DSTATE_GATHER_POOL_ALLOC

3DSTATE_GATHER_POOL_ALLOC				
Source:	RenderCS			
Length Bias:	2			
This command sets up the Gather Pool for Gather Buffers.				
Programming Notes				
If the Gather Constant Pool is enabled and RS is enabled, then for each 3DSTATE_CONSTANT_* command there must be a corresponding 3DSTATE_GATHER_CONSTANT_* command. If gather pool is enabled, then Buffer 1 of the 3DSTATE_CONSTANT command address will be an offset from the Gather Pool Base Address.				
The gather constants can only be enabled if the binding table generator is also enabled. This command must only be programmed when resource streamer is enabled to parse commands within a batch buffer.				
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:		OpCode		
23:16	3D Command Sub Opcode			
	Default Value:	1Ah 3DSTATE_GATHER_POOL_ALLOC		
	Format:	OpCode		
15:8	Reserved			
	Format:	MBZ		
7:0	DWord Length			
	Format:	=n		
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>2h</td> <td>DWORD_COUNT_n [Default]</td> </tr> </tbody> </table>	Value	Name	2h
Value	Name			
2h	DWORD_COUNT_n [Default]			
1.2	63:12	Gather Pool Base Address		
		<table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress63-12</td> </tr> </table> <p>GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p>	Format:	GraphicsAddress63-12
Format:	GraphicsAddress63-12			



3DSTATE_GATHER_POOL_ALLOC												
	11	<p>Gather Pool Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>When this bit is set it enables HW gathering of push constants. When this bit is cleared it disables HW gathering of push constants.</p>	Format:	Enable								
	Format:	Enable										
	10:7	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ											
6:0	<p>Memory Object Control State</p> <table border="1"> <tr> <td>Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for this surface.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Bit 2 is not programmable and is always zero.</td> </tr> </table>	Format:	MEMORY_OBJECT_CONTROL_STATE	Programming Notes		Bit 2 is not programmable and is always zero.						
Format:	MEMORY_OBJECT_CONTROL_STATE											
Programming Notes												
Bit 2 is not programmable and is always zero.												
3	31:12	<p>Gather Pool Buffer Size</p> <table border="1"> <tr> <td>Format:</td> <td>U20</td> </tr> </table> <p>This field specifies the size of the buffer in 4K pages. Any access which straddle or go past the end of the buffer will return undefined data. Note that BufferSize=0 indicates that there is no valid data in the buffer.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,1048575]</td> <td></td> </tr> </tbody> </table> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Restriction</th> </tr> <tr> <td colspan="2">Restriction : Programming size of zero is illegal in the case that the pool is enabled.</td> </tr> </table>	Format:	U20	Value	Name	[0,1048575]		Restriction		Restriction : Programming size of zero is illegal in the case that the pool is enabled.	
		Format:	U20									
	Value	Name										
[0,1048575]												
Restriction												
Restriction : Programming size of zero is illegal in the case that the pool is enabled.												
11:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											



3DSTATE_GS

3DSTATE_GS		
Source:		RenderCS
Length Bias:		2
Controls the GS stage hardware.		
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: 0h 3DSTATE_PIPELINED Format: OpCode
23:16	3D Command Sub Opcode	
	Default Value: 11h 3DSTATE_GS Format: OpCode	
15:8	Reserved	
7:0	Format:	MBZ
		DWord Length
	Format: =n	
1.2	63:6	Kernel Start Pointer
		Format: InstructionBaseOffset[63:6]Kernel
	This field specifies the starting location (1st GEN4 core instruction) of the kernel program run by threads spawned by this FF unit. It is specified as a 64-byte-granular offset from the Instruction Base Address.	
5:0	Reserved	
3	31	Single Program Flow
		Specifies the initial condition of the kernel program as either a single program flow (SIMDn _{xm} with m = 1) or as multiple program flows (SIMDn _{xm} with m > 1). See CR0 description in ISA Execution Environment.



3DSTATE_GS		
Value	Name	Description
0h	Disable	Single Program Flow disabled
1h	Enable	Single Program Flow enabled
30	Vector Mask Enable	
Format:		Enable Enumerated Type
Upon subsequent GS thread dispatches, this bit is loaded into the EU's Vector Mask Enable (VME, cr0.0[3]) thread state. Refer to EU documentation for the definition and use of VME state.		
Value	Name	Description
0h	Dmask	The EU will use the Dispatch Mask (supplied by the GS stage) for instruction execution.
1h	Vmask	The EU will use the Vector Mask (derived from Dispatch Mask) for instruction execution.
Programming Notes		
Under normal conditions SW shall specify DMask, as the GS stage will provide a Dispatch Mask appropriate to SIMD4x2 or SIMD8 thread execution (as a function of dispatch mode). E.g., for SIMD4x2 execution, the GS stage will generate a Dispatch Mask that is equal to what the EU would use as the Vector Mask. For SIMD8 execution there is no known usage model for use of Vector Mask (as there is for PS shaders).		
29:27	Sampler Count	
Format:		U3
Specifies how many samplers (in multiples of 4) the geometry shader kernel uses. Used only for prefetching the associated sampler state entries.		
Value	Name	Description
0h	No Samplers	No Samplers used
1h	1-4 Samplers	Between 1 and 4 samplers used
2h	5-8 Samplers	Between 5 and 8 samplers used
3h	9-12 Samplers	Between 9 and 12 samplers used
4h	13-16 Samplers	Between 13 and 16 samplers used
5h-7h	Reserved	
26	Reserved	
25:18	Binding Table Entry Count	
Format:		U8
When HW Generated Binding Table is disabled: Specifies how many binding table entries the kernel uses. Used only for prefetching of the binding table entries and associated surface state. Note: 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.		
When HW Generated Binding Table bit is enabled:		



3DSTATE_GS											
	<p>This field indicates which cache lines (512bit units - 32 Binding Table Entry section) should be fetched. Each bit in this field corresponds to a cache line. Only the 1st 4 non-zero Binding Table entries of each 32 Binding Table entry section prefetched will have its surface state prefetched.</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <th colspan="3" style="background-color: #e1eef6;">Programming Notes</th> </tr> <tr> <td colspan="3">When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.</td> </tr> </table>		Programming Notes			When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.					
Programming Notes											
When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.											
17	<p>Thread Dispatch Priority Specifies the priority of the thread for dispatch.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Normal</td> <td>Normal thread dispatch priority</td> </tr> <tr> <td>1h</td> <td>High</td> <td>High thread dispatch priority</td> </tr> </tbody> </table>		Value	Name	Description	0h	Normal	Normal thread dispatch priority	1h	High	High thread dispatch priority
Value	Name	Description									
0h	Normal	Normal thread dispatch priority									
1h	High	High thread dispatch priority									
16	<p>Floating Point Mode Specifies the initial floating point mode used by the dispatched thread.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>IEEE-754</td> <td>Use IEEE-754 Rules</td> </tr> <tr> <td>1h</td> <td>Alternate</td> <td>Use alternate rules</td> </tr> </tbody> </table>		Value	Name	Description	0h	IEEE-754	Use IEEE-754 Rules	1h	Alternate	Use alternate rules
Value	Name	Description									
0h	IEEE-754	Use IEEE-754 Rules									
1h	Alternate	Use alternate rules									
15:14	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>		Format:	MBZ							
Format:	MBZ										
13	<p>Illegal Opcode Exception Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>This bit gets loaded into EU CR0.1[12] (note the bit # difference). See <i>Exceptions and ISA Execution Environment</i>.</p>		Format:	Enable							
Format:	Enable										
12	<p>Accesses UAV</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>This field must be set when GS has a UAV access.</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <th colspan="3" style="background-color: #e1eef6;">Programming Notes</th> </tr> <tr> <td colspan="3">This field must not be set when GS Function Enable is disabled.</td> </tr> </table>		Format:	Enable	Programming Notes			This field must not be set when GS Function Enable is disabled.			
Format:	Enable										
Programming Notes											
This field must not be set when GS Function Enable is disabled.											
11	<p>Mask Stack Exception Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>This bit gets loaded into EU CR0.1[11]. See <i>Exceptions and ISA Execution Environment</i>.</p>		Format:	Enable							
Format:	Enable										
10:8	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>		Format:	MBZ							
Format:	MBZ										
7	<p>Software Exception Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>This bit gets loaded into EU CR0.1[13] (note the bit # difference). See <i>Exceptions and ISA Execution Environment</i>.</p>		Format:	Enable							
Format:	Enable										



3DSTATE_GS									
	6	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
5:0	<p>Expected Vertex Count</p> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Specifies the number of vertices per input object expected by the GS thread. Input topologies not matching this expect value are discarded.</p> <p>Note that DiscardAdjacency is also considered (e.g., if the value programmed is 3 and DiscardAdjacency is set, TRILIST_ADJ and TRISTRIP_ADJ topologies are <u>not</u> discarded as they will pass 3 vertices/object to the GS threads).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[1,32]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[1,32]			
Format:	U6								
Value	Name								
[1,32]									
4.5	63:10	<p>Scratch Space Base Pointer</p> <table border="1"> <tr> <td>Format:</td> <td>GeneralStateOffset[63:10]ScratchSpace</td> </tr> </table> <p>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.</p> <p>This field is ignored if VS Function Enable is DISABLED.</p>	Format:	GeneralStateOffset[63:10]ScratchSpace					
	Format:	GeneralStateOffset[63:10]ScratchSpace							
	9:4	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ								
3:0	<p>Per-Thread Scratch Space</p> <table border="1"> <tr> <td>Format:</td> <td>U4 power of 2 Bytes over 1K Bytes</td> </tr> </table> <p>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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td></td> <td>indicating [1K Bytes, 2M Bytes]</td> </tr> </tbody> </table>	Format:	U4 power of 2 Bytes over 1K Bytes	Value	Name	Description	[0,11]		indicating [1K Bytes, 2M Bytes]
Format:	U4 power of 2 Bytes over 1K Bytes								
Value	Name	Description							
[0,11]		indicating [1K Bytes, 2M Bytes]							
6	31	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
30:29	<p>Dispatch GRF Start Register For URB Data [5:4]</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Specifies bit [5:4] of the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. The Dispatch GRF Start Register For URB Data [3:0] field is used to specify bits [3:0] of the starting GRF register number.</p>	Format:	U2						
Format:	U2								



3DSTATE_GS			
28:23	<p>Output Vertex Size</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>[0,63] indicating [1,64] 16B units</p> <p>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).</p> <p style="text-align: center;">Programming Notes</p> <p>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. 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 number of 16B units is when rendering is disabled.</p>	Format:	U6
Format:	U6		
22:17	<p>Output Topology</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">3D_Prim_Topo_Type</td> </tr> </table> <p>This field specifies the topology type (3DPrimType) to be associated with GS-thread output vertices (if any).</p>	Format:	3D_Prim_Topo_Type
Format:	3D_Prim_Topo_Type		
16:11	<p>Vertex URB Entry Read Length</p> <p>Specifies the amount of URB data read and passed in the thread payload for each Vertex URB entry, in 256-bit register increments.</p> <p style="text-align: center;">Programming Notes</p> <p>Programming Restriction: This field must be a non-zero value if Include Vertex Handles is cleared to zero.</p>		
10	<p>Include Vertex Handles</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Boolean</td> </tr> </table> <p>If set, all the input Vertex URB handles are included in the payload. These are referred to as "pull model" URB handles, as the thread will use them to read from the URB.</p> <p style="text-align: center;">Programming Notes</p> <p>Programming Restriction: This field must be set if Vertex URB Entry Read Length is cleared to zero.</p>	Format:	Boolean
Format:	Boolean		
9:4	<p>Vertex URB Entry Read Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6
Format:	U6		
3:0	<p>Dispatch GRF Start Register For URB Data</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> </table>	Format:	U4
Format:	U4		



3DSTATE_GS																				
		<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload.</td> </tr> <tr> <td colspan="2">The Dispatch GRF Start Register for URB Data [5:4] field is used to extend the range of the starting GRF register number to [0,63].</td> </tr> </tbody> </table> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0,15]</td> <td></td> <td>indicating bits [3:0] of the GRF number</td> </tr> </tbody> </table> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> If Include Vertex Handles is enabled (pull or hybrid handles case), then For simd4x2: For DUAL_OBJECT dispatch mode this field should be: $((2 * \text{numVerticesPerObject}) + 8 - 1) / 8 + 1$ For SINGLE and DUAL_INSTANCE dispatch modes this field should be: $(\text{numVerticesPerObject} + 8 - 1) / 8 + 1$ If Include Primitive ID is set, then add 1 to the value obtained by using the above </td> </tr> <tr> <td colspan="2"> If Include Vertex Handles is enabled (pull or hybrid handles case), then For simd8: For InstanceCount == 1: numVerticesPerObject 2 For InstanceCount > 1: $(\text{numVerticesPerObject} * 8 - 1) / 8 + 2$ If Include Primitive ID is set, then add 1 to the value obtained by using the above </td> </tr> </tbody> </table>	Description		Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload.		The Dispatch GRF Start Register for URB Data [5:4] field is used to extend the range of the starting GRF register number to [0,63].		Value	Name	Description	[0,15]		indicating bits [3:0] of the GRF number	Programming Notes		If Include Vertex Handles is enabled (pull or hybrid handles case), then For simd4x2: For DUAL_OBJECT dispatch mode this field should be: $((2 * \text{numVerticesPerObject}) + 8 - 1) / 8 + 1$ For SINGLE and DUAL_INSTANCE dispatch modes this field should be: $(\text{numVerticesPerObject} + 8 - 1) / 8 + 1$ If Include Primitive ID is set, then add 1 to the value obtained by using the above		If Include Vertex Handles is enabled (pull or hybrid handles case), then For simd8: For InstanceCount == 1: numVerticesPerObject 2 For InstanceCount > 1: $(\text{numVerticesPerObject} * 8 - 1) / 8 + 2$ If Include Primitive ID is set, then add 1 to the value obtained by using the above	
Description																				
Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload.																				
The Dispatch GRF Start Register for URB Data [5:4] field is used to extend the range of the starting GRF register number to [0,63].																				
Value	Name	Description																		
[0,15]		indicating bits [3:0] of the GRF number																		
Programming Notes																				
If Include Vertex Handles is enabled (pull or hybrid handles case), then For simd4x2: For DUAL_OBJECT dispatch mode this field should be: $((2 * \text{numVerticesPerObject}) + 8 - 1) / 8 + 1$ For SINGLE and DUAL_INSTANCE dispatch modes this field should be: $(\text{numVerticesPerObject} + 8 - 1) / 8 + 1$ If Include Primitive ID is set, then add 1 to the value obtained by using the above																				
If Include Vertex Handles is enabled (pull or hybrid handles case), then For simd8: For InstanceCount == 1: numVerticesPerObject 2 For InstanceCount > 1: $(\text{numVerticesPerObject} * 8 - 1) / 8 + 2$ If Include Primitive ID is set, then add 1 to the value obtained by using the above																				
7	31:26	Reserved <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
	Format:	MBZ																		
	25:24	Reserved <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																			
23:20	Control Data Header Size <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 60%;">Format:</td> <td>U4</td> </tr> </table> <p>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 maximum number of vertices output by the GS thread. It is UNDEFINED for a GS thread to report more output vertices than can be accommodated in a non-zero-sized header. (If the header size is zero, by definition neither cut nor StreamID bits are defined.)</p> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 40%;">Value</th> <th style="width: 60%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,8]</td> <td>32B Units</td> </tr> </tbody> </table>	Format:	U4	Value	Name	[0,8]	32B Units													
Format:	U4																			
Value	Name																			
[0,8]	32B Units																			



3DSTATE_GS

19:15	Instance Control		
Format:		U5-1 #Instances	
<p>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].</p> <p>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.</p> <p>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.</p>			
Value	Name	Description	
[0,31]		Indicating [1,31] instances	
14:13	Default Stream Id		
Format:		U2	
<p>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.</p>			
12:11	Dispatch Mode		
Format:		U2	
<p>This field specifies how the GS unit dispatches multiple instances and/or multiple objects.</p>			
Value	Name	Description	Programming Notes
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.	
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).	



3DSTATE_GS			
	3h	SIMD8	Each thread shades up to 8 different objects or (if InstanceCount > 1) 8 instances of a single object. The driver must send pipe control with a cs stall after a 3dstate_gs state change and the Dispatch Mode is simd8 and the number of handles allocated to gs is less than 16.
Programming Notes			
The GS must be allocated at least two URB handles or behavior is UNDEFINED for Dual Instance or Dual Object mode.			
10	Statistics Enable		
Format:		Enable	
This bit controls whether GS-unit-specific statistics register(s) can be incremented.			
	Value	Name	Description
	0h	Disable	GS_INVOCATIONS_COUNT and GS_PRIMITIVES_COUNT cannot increment
	1h	Enable	GS_INVOCATIONS_COUNT and GS_PRIMITIVES_COUNT can increment
9:5	Invocations Increment Value		
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 incremented by twice this value.			
	Value	Name	Description
	[0,31]		indicating an increment of [1,32]
4	Include Primitive ID		
Format:		Boolean	
If set, R1 of the payload is written with Primitive ID value(s). If clear, these Primitive ID values are not included in the payload R1.			
3	Hint		
Format:		U1	
This state bit is simply passed in GS thread payloads for use by the GS kernel - it has no other impact on hardware operation.			



3DSTATE_GS											
2	<p>Reorder Mode This bit controls how vertices of triangle objects resulting from TRISTRIP[_ADJ][_REV] topologies are [re]ordered when passed in the GS thread payload See Object Vertex Ordering table (below).</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">LEADING</td> <td>Reorder the vertices of alternating triangles of a TRISTRIP[_ADJ] such that the leading (first) vertices are in consecutive order starting at v0. A similar reordering is performed on alternating triangles in a TRISTRIP_REV.</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">TRAILING</td> <td>Reorder 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.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Workaround: reorder mode must be set to REORDER_LEADING and reordering must be done in the Geometry shader.</p>		Value	Name	Description	0h	LEADING	Reorder the vertices of alternating triangles of a TRISTRIP[_ADJ] such that the leading (first) vertices are in consecutive order starting at v0. A similar reordering is performed on alternating triangles in a TRISTRIP_REV.	1h	TRAILING	Reorder 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.
Value	Name	Description									
0h	LEADING	Reorder the vertices of alternating triangles of a TRISTRIP[_ADJ] such that the leading (first) vertices are in consecutive order starting at v0. A similar reordering is performed on alternating triangles in a TRISTRIP_REV.									
1h	TRAILING	Reorder 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.									
1	<p>Discard Adjacency</p> <table border="1"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;">Enable</td> </tr> </table> <p>When set, adjacent vertices <u>will not be passed</u> in the GS payload when objects with adjacency are processed. Instead, only the non-adjacent vertices will be passed in the same fashion as the without-adjacency form of the primitive. Software should set this bit whenever a GS kernel is used that <u>does not expect</u> adjacent vertices. This allows both with-adjacency/without-adjacency variants of the primitive to be submitted to the pipeline (via 3DPRIMITIVE) - the GS unit will silently discard any adjacent vertices and present the GS thread with only the internal object. When clear, adjacent vertices <u>will be passed</u> to the GS thread, as dictated by the incoming primitive type. Software should only clear this bit when a GS kernel is used that does expect adjacent vertices. E.g., if the GS kernel is compiled to expect a TRIANGLE_ADJ object, software must clear this bit. Software should also clear this bit if the GS kernel expects a POINT or PATCHLIST_n object (which don't have with-adjacency variants).</p> <p>The only hardware assistance is to allow the submission of a with-adjacency variant of a primitive when operating with a GS kernel that expects the without-adjacency variant of the object. (E.g., when the GS kernel is compiled to expect a TRIANGLE object, 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 LINESSTRIP_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.</p>		Format:	Enable							
Format:	Enable										
0	<p>Enable</p> <table border="1"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;">Enable</td> </tr> </table> <p>Specifies whether the GS stage is enabled or disabled (pass-through).</p>		Format:	Enable							
Format:	Enable										



3DSTATE_GS													
8	31	<p>Control Data Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>This field specifies the format of the control data header (if any).</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>CUT</td> <td>The control data header contains cut bits.</td> </tr> <tr> <td>1h</td> <td>SID</td> <td>The control data header contains StreamID bits. . Output Topology must be set to POINTLIST, or behavior is UNDEFINED.</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	0h	CUT	The control data header contains cut bits.	1h	SID	The control data header contains StreamID bits. . Output Topology must be set to POINTLIST, or behavior is UNDEFINED.
	Format:	U1											
	Value	Name	Description										
	0h	CUT	The control data header contains cut bits.										
	1h	SID	The control data header contains StreamID bits. . Output Topology must be set to POINTLIST, or behavior is UNDEFINED.										
	30	<p>Static Output</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>Specifies whether the GS shader outputs a static number of vertices per invocation. If this bit is clear, the number of vertices output by each GS shader invocation is stored by the GS thread at the very beginning of the output URB entry (see GS URB Entry section below).</p>	Format:	Enable									
	Format:	Enable											
	29:27	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												
26:16	<p>Static Output Vertex Count</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">U11 Count of object vertices</td> </tr> </table> <p>If GSEnable is set and StaticOutput is set, this field specifies the total number of vertices output each GS shader invocation. If GSEnable is set and StaticOutput is clear (variable GS output), the total number of vertices output by a GS shader invocation is stored by the thread at the very beginning of the output URB entry. This field is then ignored. (See GS URB Entry below).</p>	Format:	U11 Count of object vertices										
Format:	U11 Count of object vertices												
15:9	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
8:0	<p>Maximum Number of Threads</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">U9-1 Thread count</td> </tr> </table> <p>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 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.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>[7,335]</td> <td></td> <td>indicating thread count of [8,336]</td> </tr> </tbody> </table>	Format:	U9-1 Thread count	Value	Name	Description	[7,335]		indicating thread count of [8,336]				
Format:	U9-1 Thread count												
Value	Name	Description											
[7,335]		indicating thread count of [8,336]											
9	31:28	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ											
27	<p>Reserved</p>												



3DSTATE_GS									
26:21	<p>Vertex URB Entry Output Read Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U6</td> </tr> </table> <p>Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB by SBE.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,63]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,63]			
Format:	U6								
Value	Name								
[0,63]									
20:16	<p>Vertex URB Entry Output Length</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U5</td> </tr> </table> <p>Specifies the amount of URB data written for each Vertex URB entry, in 256-bit register increments.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[1,16]</td> <td></td> </tr> </tbody> </table> <table border="1" style="width: 100%; text-align: center; margin-top: 10px;"> <tr> <td>Programming Notes</td> </tr> <tr> <td>This length does not include the vertex header.</td> </tr> </table>	Format:	U5	Value	Name	[1,16]		Programming Notes	This length does not include the vertex header.
Format:	U5								
Value	Name								
[1,16]									
Programming Notes									
This length does not include the vertex header.									
15:8	<p>User Clip Distance Clip Test Enable Bitmask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable[8]</td> </tr> </table> <p>This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept / must clip determination needs to be made. DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.</p>	Format:	Enable[8]						
Format:	Enable[8]								
7:0	<p>User Clip Distance Cull Test Enable Bitmask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable[8]</td> </tr> </table> <p>This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept determination needs to be made (does not cause a must clip). DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.</p>	Format:	Enable[8]						
Format:	Enable[8]								



3DSTATE_HIER_DEPTH_BUFFER

3DSTATE_HIER_DEPTH_BUFFER				
Source:	RenderCS			
Length Bias:	2			
Description				
<p>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).</p> <p>WM HW will internally manage the draining pipe and flushing of the caches when this command is issued. The PIPE_CONTROL restrictions are removed.</p>				
Programming Notes				
<p>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).</p>				
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: 0h 3DSTATE_PIPELINED		
	Format: OpCode			
	23:16	3D Command Sub Opcode		
		Default Value: 07h 3DSTATE_HIER_DEPTH_BUFFER		
	Format: OpCode			
15:8	Reserved			
	Format: MBZ			
7:0	Dword Length			
	Format: =n Total Length - 2			
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>3h</td> <td>Excludes Dword (0,1) [Default]</td> </tr> </tbody> </table>	Value	Name	3h
Value	Name			
3h	Excludes Dword (0,1) [Default]			



3DSTATE_HIER_DEPTH_BUFFER									
1	31:25	<p>Hierarchical Depth Buffer Object Control State</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for the hierarchical depth buffer.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE					
	Format:	MEMORY_OBJECT_CONTROL_STATE							
	24:23	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
	22	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ								
21:17	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ								
16:0	<p>Surface Pitch</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>U17-1 Pitch in Bytes</td> </tr> </table> <p>This field specifies the pitch of the hierarchical depth buffer in (#Bytes - 1).</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[127, 1FFFFh]</td> <td>corresponding to [128B, 128KB] also restricted to a multiple of 128B</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Since this surface is tiled, the pitch specified must be a multiple of the tile pitch, in the range [128B, 128KB].</p>	Format:	U17-1 Pitch in Bytes	Value	Name	[127, 1FFFFh]	corresponding to [128B, 128KB] also restricted to a multiple of 128B		
Format:	U17-1 Pitch in Bytes								
Value	Name								
[127, 1FFFFh]	corresponding to [128B, 128KB] also restricted to a multiple of 128B								
2..3	63:0	<p>Surface Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>GraphicsAddress[63:0]HierarchicalDepthBuffer</td> </tr> </table> <p>This field specifies the address of the buffer in Graphics Memory.</p> <p style="text-align: center;">Programming Notes</p> <p>The Hierarchical Depth Buffer can only be mapped to Main Memory (uncached).</p>	Format:	GraphicsAddress[63:0]HierarchicalDepthBuffer					
Format:	GraphicsAddress[63:0]HierarchicalDepthBuffer								
4	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
14:0	<p>Surface QPitch</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>QPitch[16:2]</td> </tr> </table> <p style="text-align: center;">Description</p> <p>The interpretation of this field is dependent on Surface Type as follows:</p> <ul style="list-style-type: none"> • SURFTYPE_1D: distance in <i>pixels</i> between array slices • SURFTYPE_2D/CUBE: distance in <i>rows</i> between array slices • SURFTYPE_3D: distance in <i>rows</i> between R-slices <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[4h, 1FFFCh]</td> <td></td> <td>in multiples of 4 (low 2 bits missing)</td> </tr> </tbody> </table>	Format:	QPitch[16:2]	Value	Name	Description	[4h, 1FFFCh]		in multiples of 4 (low 2 bits missing)
Format:	QPitch[16:2]								
Value	Name	Description							
[4h, 1FFFCh]		in multiples of 4 (low 2 bits missing)							



3DSTATE_HIER_DEPTH_BUFFER

Programming Notes

This field must be set to an integer multiple of 8 (QPitch[2] MBZ) Software must ensure that this field is set to a value sufficiently large such that the array slices in the surface do not overlap. Refer to the Memory Data Formats section for information on how surfaces are stored in memory.



3DSTATE_HS

3DSTATE_HS														
Source:		RenderCS												
Length Bias:		2												
Controls the HS stage hardware.														
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:	0h 3DSTATE_PIPELINED											
		Format:	OpCode											
	23:16	3D Command Sub Opcode												
Default Value:		1Bh 3DSTATE_HS												
Format:		OpCode												
15:8	Reserved													
	Format:	MBZ												
7:0	DWord Length	Format:	=n											
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>	Value	Name	7	Excludes DWord (0,1) [Default]								
	Value	Name												
	7	Excludes DWord (0,1) [Default]												
1	31:30	Reserved												
		Format:	MBZ											
1	29:27	Sampler Count												
		Format:	U3											
		Specifies how many samplers (in multiples of 4) the HS kernels use. Used only for prefetching the associated sampler state entries.												
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 55%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No Samplers</td> <td>no samplers used</td> </tr> <tr> <td>1h</td> <td>1-4 Samplers</td> <td>between 1 and 4 samplers used</td> </tr> <tr> <td>2h</td> <td>5-8 Samplers</td> <td>between 5 and 8 samplers used</td> </tr> </tbody> </table>	Value	Name	Description	0h	No Samplers	no samplers used	1h	1-4 Samplers	between 1 and 4 samplers used	2h	5-8 Samplers	between 5 and 8 samplers used
		Value	Name	Description										
		0h	No Samplers	no samplers used										
1h	1-4 Samplers	between 1 and 4 samplers used												
2h	5-8 Samplers	between 5 and 8 samplers used												



3DSTATE_HS		
	3h	9-12 Samplers between 9 and 12 samplers used
	4h	13-16 Samplers between 13 and 16 samplers used
	5h-7h	Reserved Reserved
26	Reserved	
	Format:	MBZ
25:18	Binding Table Entry Count	
	Format:	U8
	When HW Generated Binding Table is disabled: Specifies how many binding table entries the kernel uses. Used only for prefetching of the binding table entries and associated surface state. Note: 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.	
	Programming Notes	
	When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.	
17	Thread Dispatch Priority	
	Specifies the priority of the thread for dispatch	
	Value	Name Description
	0h	Normal Normal Priority
	1h	High High Priority
16	Floating Point Mode	
	Specifies the initial floating point mode used by the dispatched thread.	
	Value	Name Description
	0h	IEEE-754 Use IEEE-754 Rules
	1h	alternate Use alternate rules
15:14	Reserved	
	Format:	MBZ
13	Illegal Opcode Exception Enable	
	Format:	Enable
	This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.	
12	Software Exception Enable	
	Format:	Enable
	This bit gets loaded into EU CRO1[13] (note the bit # difference). See Exceptions and ISA Execution Environment.	
11:8	Reserved	
	Format:	MBZ



3DSTATE_HS										
	7:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ									
2	31	<p>Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>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 kernel is not expecting. E.g., if the HS kernel is expecting PATCHLIST_32 topologies, MI_TOPOLOGY_FILTER must be set to PATCHLIST_32 so only those topologies can reach the enabled HS.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th colspan="3" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="3">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.</td> </tr> </table>	Format:	Enable	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.		
	Format:	Enable								
	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	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
	Format:	MBZ								
	29	<p>Statistics Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit controls whether HS-unit-specific statistics register(s) will increment (for each patch).</p>	Format:	Enable						
	Format:	Enable								
28:27	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
26:18	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
17	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
16:8	<p>Maximum Number of Threads</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U9-1</td> </tr> </table> <p>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 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.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0,335]</td> <td></td> <td>indicating thread count of [1,336]</td> </tr> </tbody> </table>	Format:	U9-1	Value	Name	Description	[0,335]		indicating thread count of [1,336]	
Format:	U9-1									
Value	Name	Description								
[0,335]		indicating thread count of [1,336]								
7:5	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
4	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									



3DSTATE_HS										
	3:0	<p>Instance Count</p> <table border="1"> <tr> <td>Format:</td> <td>U4-1</td> </tr> </table> <p>This field determines the number of threads (minus one) spawned per input patch. If the HS kernel uses a barrier function, software must restrict the Instance Count to the number of threads that can be simultaneously active within a subslice. Factors which must be considered includes scratch memory availability.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,15]</td> <td></td> <td>representing [1,16] instances</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Instance Count must be programmed to 0 (1 instance) whenever DispatchMode is programmed to DUAL_PATCH.</p>	Format:	U4-1	Value	Name	Description	[0,15]		representing [1,16] instances
	Format:	U4-1								
	Value	Name	Description							
	[0,15]		representing [1,16] instances							
3.4	63:6	<p>Kernel Start Pointer</p> <table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[63:6]Kernel</td> </tr> </table> <p>This field specifies the starting location (1st GEN core instruction) of the kernel program run by threads spawned by this FF unit. It is specified as a 64-byte-granular offset from the Instruction Base Address.</p>	Format:	InstructionBaseOffset[63:6]Kernel						
	Format:	InstructionBaseOffset[63:6]Kernel								
5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
5..6	63:10	<p>Scratch Space Base Pointer</p> <table border="1"> <tr> <td>Format:</td> <td>GeneralStateOffset[63:10]</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td></td> <td>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 this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space.</td> </tr> </tbody> </table>	Format:	GeneralStateOffset[63:10]	Value	Name	Description	[0,31]		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 this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space.
		Format:	GeneralStateOffset[63:10]							
	Value	Name	Description							
	[0,31]		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 this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space.							
9:4	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
3:0	<p>Per-Thread Scratch Space</p> <table border="1"> <tr> <td>Format:</td> <td>U4 power of 2 Bytes over 1K Bytes</td> </tr> </table> <p>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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td></td> <td>Indicating[1K Bytes, 2M Bytes</td> </tr> </tbody> </table>	Format:	U4 power of 2 Bytes over 1K Bytes	Value	Name	Description	[0,11]		Indicating[1K Bytes, 2M Bytes	
	Format:	U4 power of 2 Bytes over 1K Bytes								
Value	Name	Description								
[0,11]		Indicating[1K Bytes, 2M Bytes								



3DSTATE_HS												
7	31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	28	<p>Dispatch GRF Start Register For URB Data [5]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Specifies bit [5] of the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. The Dispatch GRF Start Register For URB Data [4:0] field is used to specify bits [4:0] of the starting GRF register number.</p>	Format:	U1								
	Format:	U1										
27	<p>Single Program Flow</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>Specifies the initial condition of the kernel program as either a single program flow (SIMDn_{xm} with m = 1) or as multiple program flows (SIMDn_{xm} with m > 1). See CR0 description in <i>ISA Execution Environment</i>.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Single Program Flow Enabled</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0h	Reserved		1h	Enable	Single Program Flow Enabled
Format:	Enable											
Value	Name	Description										
0h	Reserved											
1h	Enable	Single Program Flow Enabled										
26	<p>Vector Mask Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">U1 Enumerated Type</td> </tr> </table> <p>Upon subsequent HS thread dispatches, this bit is loaded into the EU's Vector Mask Enable (VME, cr0.0[3]) thread state. Refer to the EU documentation for the definition and use of VME state.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 75%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Dmask</td> <td>The EU will use the Dispatch Mask (supplied by the HS stage) for instruction execution.</td> </tr> <tr> <td>1h</td> <td>Vmask</td> <td>The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Under normal conditions SW shall specify DMask, as the HS stage will provide a Dispatch Mask appropriate to SIMD4x2 or SIMD8 thread execution (as a function of dispatch mode). E.g., for SIMD4x2 thread execution, the HS state will generate a Dispatch Mask that is equal to what the EU would use as a Vector Mask. For SIMD8 execution there is no known usage model for use of Vector Mask (as there is for PS shaders).</p>	Format:	U1 Enumerated Type	Value	Name	Description	0h	Dmask	The EU will use the Dispatch Mask (supplied by the HS stage) for instruction execution.	1h	Vmask	The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.
Format:	U1 Enumerated Type											
Value	Name	Description										
0h	Dmask	The EU will use the Dispatch Mask (supplied by the HS stage) for instruction execution.										
1h	Vmask	The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.										
25	<p>Accesses UAV</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>This field must be set when HS has a UAV access</p> <p style="text-align: center;">Programming Notes</p> <p>This field must not be set when HS Function Enable is disabled.</p>	Format:	Enable									
Format:	Enable											



3DSTATE_HS														
24	Include Vertex Handles													
	Format:	Boolean												
<p>If set, all the input Vertex URB handles are included in payloads. This field is ignored if HS Function Enable is DISABLED.</p>														
Programming Notes														
<p>Programming Restriction: This field must be set if value if Vertex URB Entry Read Length is cleared to zero.</p>														
23:19	Dispatch GRF Start Register For URB Data													
	Format:	U5												
	Description													
	<p>Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. This field is ignored if HS Function Enable is DISABLED.</p>													
	<p>The Dispatch GRF Start Register for URB Data [5] field is used to extend the range of the starting GRF register number to [0,63].</p>													
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td></td> <td>indicating bits [4:0] of the GRF number</td> </tr> </tbody> </table>		Value	Name	Description	[0,31]		indicating bits [4:0] of the GRF number						
Value	Name	Description												
[0,31]		indicating bits [4:0] of the GRF number												
Programming Notes														
<p>When Include Vertex Handles is set for non-instanced 8_PATCH dispatch of PATCHLIST_30..32 objects, pushed vertex data and/or pushed constants cannot be used as they would need to start in the payload beyond the range of this field (i.e., beyond R31). When Include PrimitiveID is also set, this issue extends to non-instanced 8_PATCH dispatch of PATCHLIST_29..32 objects.</p>														
18:17	Dispatch Mode													
	Format:	U2												
	<p>This field is unused to set the current thread dispatch mode for the HS stage.</p>													
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>SINGLE_PATCH</td> <td>HS threads are passed inputs and an output handle associated with a single input patch.</td> </tr> <tr> <td>1h</td> <td>DUAL_PATCH</td> <td>HS threads are passed inputs and an output handle associated with (up to) two input patches. Patch 0 data is passed in the four lower channels while Patch 1 data (if present) is passed in the four upper channels. Restrictions: Only valid for 4 or fewer input control points (PATCHLIST_4 and below). SW is expected to use MI_TOPOLOGY_FILTER to ensure only patches with the expected # of ICPs are processed by the pipeline when HS is enabled.</td> </tr> <tr> <td>2h</td> <td>8_PATCH</td> <td>HS threads are passed inputs and an output handle associated with (up to) 8 patches in SIMD8 fashion. See 8_PATCH Thread Payload.</td> </tr> </tbody> </table>		Value	Name	Description	0h	SINGLE_PATCH	HS threads are passed inputs and an output handle associated with a single input patch.	1h	DUAL_PATCH	HS threads are passed inputs and an output handle associated with (up to) two input patches. Patch 0 data is passed in the four lower channels while Patch 1 data (if present) is passed in the four upper channels. Restrictions: Only valid for 4 or fewer input control points (PATCHLIST_4 and below). SW is expected to use MI_TOPOLOGY_FILTER to ensure only patches with the expected # of ICPs are processed by the pipeline when HS is enabled.	2h	8_PATCH	HS threads are passed inputs and an output handle associated with (up to) 8 patches in SIMD8 fashion. See 8_PATCH Thread Payload.
	Value	Name	Description											
0h	SINGLE_PATCH	HS threads are passed inputs and an output handle associated with a single input patch.												
1h	DUAL_PATCH	HS threads are passed inputs and an output handle associated with (up to) two input patches. Patch 0 data is passed in the four lower channels while Patch 1 data (if present) is passed in the four upper channels. Restrictions: Only valid for 4 or fewer input control points (PATCHLIST_4 and below). SW is expected to use MI_TOPOLOGY_FILTER to ensure only patches with the expected # of ICPs are processed by the pipeline when HS is enabled.												
2h	8_PATCH	HS threads are passed inputs and an output handle associated with (up to) 8 patches in SIMD8 fashion. See 8_PATCH Thread Payload.												



3DSTATE_HS		
	3h	Reserved
16:11	Vertex URB Entry Read Length	
	Format:	U6
	Specifies the amount of URB data read and passed in the thread payload for each Vertex URB entry, in 256-bit register increments. This field is ignored if HS Function Enable is DISABLED.	
	Value	Name
	[0,63]	
	Programming Notes	
	Programming Restriction: This field must be a non-zero value if Include Vertex Handles is cleared to zero.	
10	Reserved	
	Format:	MBZ
9:4	Vertex URB Entry Read Offset	
	Format:	U6
	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. This field is ignored if HS Function Enable is DISABLED.	
	Value	Name
	[0,63]	
3:1	Reserved	
	Format:	MBZ
0	Include Primitive ID	
	Format:	Enable
	If set, R1 of the payload is written with Primitive ID value(s). If clear, these Primitive IDs are not included in the payload R1.	
	Programming Notes	
	This field is only used when DUAL_PATCH DispatchMode is specified. In SINGLE_PATCH, the single Primitive ID is always passed in R0.	
8	31:0	Reserved
	Format:	MBZ



3DSTATE_INDEX_BUFFER

3DSTATE_INDEX_BUFFER			
Source:	RenderCS		
Length Bias:	2		
<p>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.</p>			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		0Ah 3DSTATE_INDEX_BUFFER	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	3h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:11	Reserved	
		Format:	MBZ
	10	Reserved	
9:8	9:8	Index Format	
		Format:	U2 Enumerated type
		This field specifies the data format of the index buffer. All index values are UNSIGNED.	
		Value	Name
		0h	BYTE



3DSTATE_INDEX_BUFFER							
	<table border="1"> <tr> <td>1h</td> <td>WORD</td> </tr> <tr> <td>2h</td> <td>DWORD</td> </tr> </table>	1h	WORD	2h	DWORD		
1h	WORD						
2h	DWORD						
7	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
6:0	<p>Memory Object Control State</p> <table border="1"> <tr> <td>Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for this index buffer.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE				
Format:	MEMORY_OBJECT_CONTROL_STATE						
2..3	<p>63:0 Buffer Starting Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[63:0]Index_Buffer_Entry</td> </tr> </table> <p>This field contains the size-aligned (as specified by Index Format) Graphics Address LSBs of the first element of interest within the index buffer. Software must program this value with the combination (sum) of the base address of the memory resource and the byte offset from the base address to the starting structure within the buffer.</p> <table border="1" style="background-color: #e6f2ff;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Index Buffers can only be allocated in linear (not tiled) graphics memory.</td> </tr> </table>	Format:	GraphicsAddress[63:0]Index_Buffer_Entry	Programming Notes		Index Buffers can only be allocated in linear (not tiled) graphics memory.	
Format:	GraphicsAddress[63:0]Index_Buffer_Entry						
Programming Notes							
Index Buffers can only be allocated in linear (not tiled) graphics memory.							
4	<p>31:0 Buffer Size</p> <table border="1"> <tr> <td>Format:</td> <td>U32 Count of bytes</td> </tr> </table> <p>This field specifies the size of the buffer in bytes. Index accesses which straddle or go past the end of the buffer will return 0..Note that BufferSize=0 indicates that there is no valid data in the buffer.</p> <table border="1" style="background-color: #e6f2ff;"> <thead> <tr> <th style="width: 60%;">Value</th> <th style="width: 40%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, FFFFFFFFh]</td> <td></td> </tr> </tbody> </table>	Format:	U32 Count of bytes	Value	Name	[0, FFFFFFFFh]	
Format:	U32 Count of bytes						
Value	Name						
[0, FFFFFFFFh]							



3DSTATE_LINE_STIPPLE

3DSTATE_LINE_STIPPLE			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_LINE_STIPPLE command is used to specify state variables used in the Line Stipple function.			
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:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	08h 3DSTATE_LINE_STIPPLE	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	Dword Length		
	Default Value:	1h Excludes Dword (0,1)	
	Format:	=n Total Length - 2	
1	31	Modify Enable (Current Repeat Counter, Current Stipple Index)	
		Format:	Enable
	Modify enable for Current Repeat Counter and Current Stipple Index fields.		
	<p style="text-align: center;">Programming Notes</p> <p>It is provided only for HW-generated commands as part of context save/restore. SW must initialize the current repeat counter, current stipple count fields if it sets this bit to enable. SW must set this bit to reset the stipple count.</p>		
30	Reserved		
	Format:	MBZ	
29:21	Current Repeat Counter		
	Format:	U9	
This field sets the HW-internal repeat counter state. SW must initialize it to 1 if the modify enable is set.			



3DSTATE_LINE_STIPPLE							
	20	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ					
	19:16	<p>Current Stipple Index</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field sets the HW-internal stipple pattern index. SW must initialize it to 0 if the modify enable is set.</p>	Format:	U4			
Format:	U4						
15:0	<p>Line Stipple Pattern</p> <table border="1"> <tr> <td>Format:</td> <td>16 bit mask Bit 15 = most significant bit, Bit 0 = least significant bit</td> </tr> </table> <p>Specifies a pattern used to mask out bit specific pixels while rendering lines.</p>	Format:	16 bit mask Bit 15 = most significant bit, Bit 0 = least significant bit				
Format:	16 bit mask Bit 15 = most significant bit, Bit 0 = least significant bit						
2	31:15	<p>Line Stipple Inverse Repeat Count</p> <table border="1"> <tr> <td>Format:</td> <td>U1.16</td> </tr> </table> <p>Range: [0.00390625, 1.0]</p> <p>Specifies the inverse (truncated) of the repeat count for the line stipple function.</p>	Format:	U1.16			
	Format:	U1.16					
	14:9	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ					
8:0	<p>Line Stipple Repeat Count</p> <table border="1"> <tr> <td>Format:</td> <td>U9</td> </tr> </table> <p>Specifies the repeat count for the line stipple function.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[1, 256]</td> <td></td> </tr> </tbody> </table>	Format:	U9	Value	Name	[1, 256]	
Format:	U9						
Value	Name						
[1, 256]							



3DSTATE_MONOFILTER_SIZE

3DSTATE_MONOFILTER_SIZE			
Source:		RenderCS	
Length Bias:		2	
This state specifies the size of the filter which is used when filtering in MAPFILTER_MONO mode.			
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:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	11h 3DSTATE_MONOFILTER_SIZE	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31:6	Reserved	
		Format:	MBZ
	5:3	Monochrome Filter Width	
Format:		U3	
This field specifies the width of the monochrome filter. It is ignored if the monochrome filter is not enabled.			
		Value	
		Name	
[1,7]			



3DSTATE_MONOFILTER_SIZE							
2:0	<p>Monochrome Filter Height</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>This field specifies the height of the monochrome filter. It is ignored if the monochrome filter is not enabled.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[1,7]</td> <td></td> </tr> </tbody> </table>	Format:	U3	Value	Name	[1,7]	
Format:	U3						
Value	Name						
[1,7]							



3DSTATE_MULTISAMPLE

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.			
Programming Notes			
It is illegal to render to surfaces with multiple different values of the state fields in this command.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	0Dh 3DSTATE_MULTISAMPLE	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:6	Reserved	
		Format:	MBZ
5	Pixel Position Offset Enable	Format:	Enable
		Enables the device to offset pixel positions by 0.5 both in horizontal and vertical directions.	
	Programming Notes		
	Setting this field along with setting the Pixel Location to upper left and number of multisamples to greater than one will cause the device to offset pixel positions by 0.5 both in horizontal and vertical directions.		



3DSTATE_MULTISAMPLE

It is to be noted this is done to adjust the pixel co-ordinate system to DX9 like, so any WM_HZ_OP screen space rectangles (eg: legacy HiZ Clear, Resolve etc) generated internally by driver in this mode needs to be aware of this offset adjustment and send the rectangles according to alignment restriction taking this offset adjustment into consideration.
SW can choose to set this bit only for DX9 API. DX10/OGL API's should not have any effect by setting or not setting this bit.

4 Pixel Location

Format:	U1
---------	----

This field specifies where the device evaluates "pixel" (vs. centroid or sample) values/attributes.

Value	Name	Description
0h	CENTER	Use the pixel center (0.5, 0.5 offset)
1h	UL_CORNER	Use the pixel upper-left corner

Programming Notes

The programming of this field is assumed to be a function of the API being supported. Specifically, it is expected that OpenGL and DX10+ APIs require CENTER selection, while DX9-APIs require UL_CORNER selection.

When 3DSTATE_RASTER::**ForcedSampleCount** is other than NUMRASTSAMPLES_0, this field must be 0h.

3:1 Number of Multisamples

Format:	U3
---------	----

This field specifies how many samples/pixel exist in all RTs and the Depth Buffer, as $\log_2(\#samples)$. This field is valid regardless of the setting of **Multisample Rasterization Mode**.

Value	Name	Description
0h	1	1 sample/pixel
1h	2	2 samples/pixel
2h	4	4 samples/pixel
3h	8	8 samples/pixel
4h	16	16 samples/pixel
5h-7h	Reserved	

Programming Notes

The setting of this field must match the **Number of Multisamples** field in SURFACE_STATE of all bound render targets.

0 Reserved

Format:	MBZ
---------	-----



3DSTATE_POLY_STIPPLE_OFFSET

3DSTATE_POLY_STIPPLE_OFFSET			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_POLY_STIPPLE_OFFSET command is used to specify the origin of the repeated screen-space Polygon Stipple Pattern as an X, Y offset from the Color Buffer origin.			
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:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	06h 3DSTATE_POLY_STIPPLE_OFFSET	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	Dword Length		
	Default Value:	0h Excludes Dword (0,1)	
	Format:	=n Total Length - 2	
1	31:13	Reserved	
		Format:	MBZ
	12:8	Polygon Stipple X Offset	
		Format:	U5
		Specifies a 5 bit x address offset in the poly stipple pattern	
	Value	Name	
	[0,31]		
7:5	Reserved		
	Format:	MBZ	



3DSTATE_POLY_STIPPLE_OFFSET			
	4:0	Polygon Stipple Y Offset	
		Format: U5	
		Specifies a 5 bit y address offset in the poly stipple pattern	
		Value	Name
		[0,31]	



3DSTATE_POLY_STIPPLE_PATTERN

3DSTATE_POLY_STIPPLE_PATTERN			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_POLY_STIPPLE_PATTERN command is used to specify the 32x32 Polygon Stipple Pattern used in the Polygon Stipple function of the WM unit.			
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:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	07h 3DSTATE_POLY_STIPPLE_PATTERN	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	Dword Length		
	Default Value:	1Fh Excludes Dword (0,1)	
	Format:	=n Total Length - 2	
1..32	31:0	Pattern Row Format: 32 bit mask Bit 31 = upper left corner, Bit 0 = upper right corner of first row. Specifies a pattern used by Polygon Stipple to mask out specific pixels of every 32x32 area rendered.	



3DSTATE_PS_BLEND

3DSTATE_PS_BLEND			
Source:		RenderCS	
Length 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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		4Dh 3DSTATE_PS_BLEND	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31	Alpha To Coverage Enable	
		Format:	Enable
		If set, indicates that AlphaToCoverage is on RT[0], since this bit must be set the same for all RTs in the MRT case.	
30	Has Writeable RT		
	Format:	Enable	
		When set indicates the there is at least one non-null RT w/ at least one channel write enabled	
29	Color Buffer Blend Enable		
	Format:	Enable	
		When set indicates that RT[0] has color buffer blend enabled.	



3DSTATE_PS_BLEND			
28:24	Source Alpha Blend Factor		
	Format:	3D_Color_Buffer_Blend_Factor	
	Indicates the "source factor" in alpha Color Buffer Blending stage for RT[0]		
	23:19	Destination Alpha Blend Factor	
		Format:	3D_Color_Buffer_Blend_Factor
	Indicates the "destination factor" in alpha Color Buffer Blending stage for RT[0]		
18:14	Source Blend Factor		
	Format:	3D_Color_Buffer_Blend_Factor	
Indicates the "source factor" in Color Buffer Blending stage for RT[0]			
13:9	Destination Blend Factor		
	Format:	3D_Color_Buffer_Blend_Factor	
Indicates the "destination factor" in Color Buffer Blending stage for RT[0]			
8	Alpha Test Enable		
	Format:	Enable	
Indicates the AlphaTestEnable for RT[0]			
7	Independent Alpha Blend Enable		
	Format:	Enable	
Indicates the Independent Alpha Blend Enable for RT[0] When enabled, the other fields in this instruction control the combination of the alpha components in the Color Buffer Blend stage. When disabled, the alpha components are combined in the same fashion as the color components.			
6:0	Reserved		
	Format:	MBZ	



3DSTATE_PS

3DSTATE_PS				
Source:		RenderCS		
Length 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:		0h 3DSTATE_PIPELINED		
Format:		OpCode		
23:16	3D Command Sub Opcode			
	Default Value:	20h 3DSTATE_PS		
	Format:	OpCode		
15:8	Reserved			
	Format:	MBZ		
7:0	DWord Length			
	Default Value:	0Ah Excludes DWord (0,1)		
	Format:	=n Total Length - 2		
1..2	63:6	Kernel Start Pointer 0		
		Format: InstructionBaseOffset[63:6]Kernel Specifies the 64-byte aligned address offset of the first instruction in the kernel[0]. This pointer is relative to the Instruction Base Address .		
	5:0	Reserved		
		Format: MBZ		
3	31	Single Program Flow		
		Single Program Flow (SPF) specifies the initial condition of the kernel program as either a single program flow (SIMDn _{xm} with m = 1) or as multiple program flows (SIMDn _{xm} with m > 1). See CR0 description in ISA Execution Environment.		
		Value	Name	Description
		0h	Multiple	Multiple Program Flows
	1h	Single	Single Program Flows	



3DSTATE_PS				
30	Vector Mask Enable			
	Format: Enable			
	When SPF=0, Vector Mask Enable (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 are enabled based on the dispatch mask	
	1h	Vmask	Channels are enabled based on the vector mask	
	29:27	Sampler Count		
		Specifies how many samplers (in multiples of 4) the vertex shader 0 kernel uses. Used only for prefetching the associated sampler state entries.		
		Value	Name	Description
		[0,4]		
0h		No Samplers	no samplers used	
1h		1-4 Samplers	between 1 and 4 samplers used	
2h		5-8 Samplers	between 5 and 8 samplers used	
3h		9-12 Samplers	between 9 and 12 samplers used	
4h		13-16 Samplers	between 13 and 16 samplers used	
5h-7h			Reserved	
26	Single Precision Denormal Mode			
	Specifies the single precision denormal mode used by the dispatched thread.			
	Value	Name	Description	
	0h	Flushed to Zero	Single Precision denormals are flushed to zero	
1h	Retained	Single Precision denormals are retained		
25:18	Binding Table Entry Count			
	Description			
	Specifies how many binding table entries the kernel uses. Used only for prefetching of the binding table entries and associated surface state. Note: For kernels using a large number of binding table entries, it may be advantageous to set this field to zero to avoid prefetching too many entries and thrashing the state cache. This field is ignored if [PS Function Enable] is DISABLED.			
	When [HW Generated Binding Table] bit is enabled: This field indicates which cache lines (512bit units - 32 Binding Table Entry section) should be fetched. Each bit in this field corresponds to a cache line. Only the 1st 4 non-zero Binding Table entries of each 32 Binding Table entry section prefetched will have its surface state prefetched. See 3D Pipeline for more information.			
	Programming Notes			
When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be				



3DSTATE_PS				
	generated at JIT time.			
17	Thread Dispatch Priority Specifies the priority of the thread for dispatch.			
	Value	Name		
	0h	Normal		
	1h	High		
16	Floating Point Mode Specifies the floating point mode used by the dispatched thread.			
	Value	Name		
	0h	IEEE-754		
	1h	Alternate		
15:14	Rounding Mode Specifies the rounding mode used by the dispatched thread.			
	Value	Name		
	0h	RTNE		
	1h	RU		
	2h	RD		
	3h	RTZ		
13	Illegal Opcode Exception Enable Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>Enable</td></tr></table> This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.			Enable
	Enable			
12	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>			MBZ
	MBZ			
11	Mask Stack Exception Enable Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>Enable</td></tr></table> This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.			Enable
	Enable			
10:8	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>			MBZ
	MBZ			
7	Software Exception Enable Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>Enable</td></tr></table> This bit gets loaded into EU CR0.1[13] (note the bit # difference). See Exceptions and ISA Execution Environment.			Enable
	Enable			
6:0	Reserved			



		3DSTATE_PS	
		Format:	MBZ
4..5	63:10	Scratch Space Base Pointer	
		Format:	GeneralStateOffset[63:10]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.	
		Programming Notes	
		Scratch Space per slice is computed based on 4 sub-slices. SW must allocate scratch space enough so that each slice has 4 slices allowed.	
6	9:4	Reserved	
		Format:	MBZ
6	3:0	Per Thread Scratch 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.	
		Value	Name
		[0,11]	indicating [1k bytes, 2M bytes] in powers of two
6	31:23	Maximum Number of Threads Per PSD	
		Format:	U8-1 Representing Thread Count
		Range = [1, 63] --> [2, 64] threads. Specifies the maximum number of simultaneous threads allowed to be active per PSD.	
		With one PSD per subslice, the maximum number of PS threads in the system is this field multiplied by the number of enabled subslices in the current configuration.	
		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.	
6	22:12	Reserved	
		Format:	MBZ
		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.	
6	10	Reserved	
		Format:	MBZ
6	9	Reserved	
		Format:	MBZ



3DSTATE_PS				
8	Render Target Fast Clear Enable			
	Format:	Enable		
	This field is set to enable fast clear of the bound render targets. See "Render Target Fast Clear" for restrictions on enabling this field.			
	Programming Notes			
	When this bit is set, corresponding BTI for the render target that is being cleared must be equal to 0.			
	When this bit is set, RENDER_SURFACE_STATE type must not be NULL.			
	7:6	Render Target Resolve Type		
		Format:	U2 Enumerated Type	
		Specifies what type of Render Target Resolve is needed for the surface to be consumed properly by the end Client. Programming notes below.		
		Value	Name	Description
0h		RESOLVE_DISABLED	No Resolve Needed	
1h		RESOLVE_PARTIAL	Partial resolve is for resolving RT for clear values i.e. it leaves no cache lines at implied clear value.	Display engine does not support unresolved clear values in the display buffer, hence this resolve is required before binding any compressed RT to the display via flip commands. If non 0/1 clear values are used, software must perform partial resolve before binding cleared render target as texture.
2h		Reserved		
3h		RESOLVE_FULL	Full Resolve is for Resolving RT for Clear/Compressed to Uncompressed State	
Programming Notes				
When this bit is set, corresponding BTI for the render target that is being resolved must be equal to 0.				
When this bit is set, RENDER_SURFACE_STATE type must not be NULL.				
5	Reserved			
	Format:	MBZ		
4:3	Position XY Offset Select			
	Format:	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.				



3DSTATE_PS			
	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 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 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. However, this field can be used to pass Centroid or Sample offsets in the payload for special test modes (e.g., where barycentric coordinates are computed in the PS vs. being HW-generated and passed in the payload).			
MSDISPMODE_PERSAMPLE is required in order to select POSOFFSET_SAMPLE.			
2	32 Pixel Dispatch Enable		
	Format:	Enable	
Enables the Windower to dispatch 8 subspans in one payload. Variable Pixel Dispatch in Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.			
Programming Notes			
When NUM_MULTISAMPLES = 16 or FORCE_SAMPLE_COUNT = 16, SIMD32 Dispatch must not be enabled for PER_PIXEL dispatch mode.			
1	16 Pixel Dispatch Enable		
	Format:	Enable	
Enables the Windower to dispatch 4 subspans in one payload. Variable Pixel Dispatch in Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.			
0	8 Pixel Dispatch Enable		
	Format:	Enable	
Enables the Windower to dispatch 2 subspans from 1 object (polygon) in one payload. Variable Pixel Dispatch in Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.			
Programming Notes			
When Render Target Fast Clear Enable is ENABLED or Render Target Resolve Type = RESOLVE_PARTIAL or RESOLVE_FULL, this bit must be DISABLED.			



3DSTATE_PS						
7	31:23	Reserved Format: _____ MBZ				
	22:16	Dispatch GRF Start Register For Constant/Setup Data 0 Format: _____ U7 Specifies the starting GRF register number for the Constant/Setup portion of the thread payload for kernel[0]. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,127]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,127]	
	Value	Name				
	[0,127]					
	15	Reserved Format: _____ MBZ				
	14:8	Dispatch GRF Start Register For Constant/Setup Data 1 Format: _____ U7 Specifies the starting GRF register number for the Constant/Setup portion of the thread payload for kernel[1]. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,127]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,127]	
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]. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,127]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,127]		
Value	Name					
[0,127]						
8..9	63:6	Kernel Start Pointer 1 Format: _____ InstructionBaseOffset[63:6]Kernel Specifies the 64-byte aligned address offset of the first instruction in kernel[1]. This pointer is relative to the Instruction Base Address.				
	5:0	Reserved Format: _____ MBZ				
10..11	63:6	Kernel Start Pointer 2 Format: _____ InstructionBaseOffset[63:6]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_PS_EXTRA

3DSTATE_PS_EXTRA			
Source:		RenderCS	
Length 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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		4fh 3DSTATE_PS_EXTRA	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31	Pixel Shader Valid	
		Format:	Enable
When set indicates a valid pixel shader. When this bit clear the rest of this command should also be clear.			
1	30	Pixel Shader Does not write to RT	
		Format:	Enable
When set indicates the pixel shader does not write to render target.			
Programming Notes			
When Pixel Shader writes to UAV but does not write to RT, a dummy render target write is required to convey EOT to the PS dispatch function. Hence, this bit must be reset in this case. Whenever, there is a render target write message even to the NULL render target, this bit must be reset.			
When Pixel Shader Kills Pixel is set, SW must perform a dummy render target write from the shader and not set this bit, so that Occlusion Query is correct.			



3DSTATE_PS_EXTRA																	
29	oMask Present to Render Target																
	Format:	Enable															
<p>This bit is inserted in the PS payload header and made available to the DataPort (either via the message header or via header bypass) to indicate that oMask data from the shader (one or two phases) is included in Render Target Write messages. If present, the oMask data is used to mask off samples.</p>																	
28	Pixel Shader Kills Pixel																
	Format:	Enable															
<p>This bit, if ENABLED, indicates that the PS kernel has the ability to kill (discard) pixels or samples, other than due to depth or stencil testing. This bit is required to be ENABLED in the following situations:</p> <ul style="list-style-type: none"> The API pixel shader program contains "killpix" or "discard" instructions, or other code in the pixel shader kernel that can cause the final pixel mask to differ from the pixel mask received on dispatch. 																	
27:26	Pixel Shader Computed Depth Mode																
	Format:	U2 Enumerated Type															
<p>This field specifies the computed depth mode for the pixel shader.</p>																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>PSCDEPTH_OFF</td> <td>Pixel shader does not compute depth</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>PSCDEPTH_ON</td> <td>Pixel shader computes depth with no guarantee as to its value</td> </tr> <tr> <td style="text-align: center;">2h</td> <td>PSCDEPTH_ON_GE</td> <td></td> </tr> <tr> <td style="text-align: center;">3h</td> <td>PSCDEPTH_ON_LE</td> <td>Pixel shader computes depth and guarantees that oDepth <= SourceDepth If the Position ZW interpolation mode in 3DSTATE_WM does not match the DX Multisample Rasterization mode in 3DSTATE_RASTER, HW will internally convert to PSCDEPTH_ON.</td> </tr> </tbody> </table>			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		3h	PSCDEPTH_ON_LE	Pixel shader computes depth and guarantees that oDepth <= SourceDepth If the Position ZW interpolation mode in 3DSTATE_WM does not match the DX Multisample Rasterization mode in 3DSTATE_RASTER, HW will internally convert to PSCDEPTH_ON.
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																
3h	PSCDEPTH_ON_LE	Pixel shader computes depth and guarantees that oDepth <= SourceDepth If the Position ZW interpolation mode in 3DSTATE_WM does not match the DX Multisample Rasterization mode in 3DSTATE_RASTER, HW will internally convert to PSCDEPTH_ON.															
Programming Notes																	
<p>If this field is set to any value other than PSCDEPTH_OFF, a multi-phase shader (i.e. dispatch RATE_COARSE or RATE_PIXEL with pixel/sample loops or sample loop respectively) must output depth and render targets only at the last phase.</p>																	
25	Force Computed Depth																
	Format:	Enable															
Programming Notes																	
<p>This field should be left DISABLED. This field should not be tested for functional validation.</p>																	
24	Pixel Shader Uses Source Depth																
	Format:	Enable															
<p>This bit, if ENABLED, indicates that the PS kernel requires the source depth value (vPos.z) to be</p>																	



3DSTATE_PS_EXTRA				
	<p>passed in the payload. The source depth value is interpolated according to the Position ZW Interpolation Mode state.</p>			
23	<p>Pixel Shader Uses Source W</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the PS kernel requires the interpolated source W value (vPos.w) to be passed in the payload. The W value is interpolated according to the Position ZW Interpolation Mode state.</p>	Format:	Enable	
Format:	Enable			
22	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
21:18	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
17	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
16:11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
10	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
9	<p>Reserved</p>			
8	<p>Attribute Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>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.</p>	Format:	Enable	
Format:	Enable			
7	<p>Pixel Shader Disables Alpha To Coverage</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>When set indicates the pixel shader AlphaToCoverage should be disabled due to oMask output. The setting of this bit is API dependent.</p>	Format:	Enable	
Format:	Enable			
6	<p>Pixel Shader Is Per Sample</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit, when ENABLED, indicates that the pixel shader is dispatched at the per sample shading rate. If the bit is DISABLED, the pixel shader is dispatched at the per pixel rate.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This bit must NOT be set when PS is used to do clear MSRTs with Fast Clear Optimization Enabled.</p> <p>When NUMRASTSAMPLES is other than 1 (i.e. when FORCED_SAMPLE_COUNT is greater than 1, hence Target Independent Rasterization is enabled) , this bit MUST be DISABLED.</p>	Format:	Enable	Programming Notes
Format:	Enable			
Programming Notes				
5	<p>Pixel Shader Computes Stencil</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field when set indicates that the pixel shader computes the stencil reference value.</p>	Format:	Enable	
Format:	Enable			



3DSTATE_PS_EXTRA

Programming Notes																
If this field is ENABLED, a multi-phase shader (i.e. dispatch RATE_COARSE or RATE_PIXEL with pixel/sample loops or sample loop respectively) must output stencil and render targets only at the last phase.																
4	Reserved Format: MBZ															
3	Pixel Shader Pulls Bary Format: Enable This bit indicates if Pixel Shader uses Pull Bary i.e. PI message. If this bit is reset, PS does not do Pull Bary.															
2	Pixel Shader Has UAV Format: Enable Format: U1 Enumerated Type This field when set indicates that the pixel shader has a UAV attached to it.															
1:0	Input Coverage Mask State Format: U2 This field indicates the type of input coverage mask that the PS kernel requires to be passed in the payload. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NONE</td> <td>Pixel shader does not use input coverage masks.</td> </tr> <tr> <td>1h</td> <td>NORMAL</td> <td>Input Coverage masks based on outer conservatism and factors in SAMPLE_MASKs. If Pixel is conservatively covered, all samples are enabled.</td> </tr> <tr> <td>2h</td> <td>INNER_CONSERVATIVE</td> <td>Input Coverage masks based on inner conservatism and factors in SAMPLE_MASKs. If Pixel is conservatively fully covered all samples are enabled.</td> </tr> <tr> <td>3h</td> <td>DEPTH_COVERAGE</td> <td>Input coverage masks are computed after factoring depth/stencil test results, only if Early Depth Stencil Test is enabled. If Early Depth Stencil Test is not enabled, HW uses NORMAL Input Coverage Masks.</td> </tr> </tbody> </table>	Value	Name	Description	0h	NONE	Pixel shader does not use input coverage masks.	1h	NORMAL	Input Coverage masks based on outer conservatism and factors in SAMPLE_MASKs. If Pixel is conservatively covered, all samples are enabled.	2h	INNER_CONSERVATIVE	Input Coverage masks based on inner conservatism and factors in SAMPLE_MASKs. If Pixel is conservatively fully covered all samples are enabled.	3h	DEPTH_COVERAGE	Input coverage masks are computed after factoring depth/stencil test results, only if Early Depth Stencil Test is enabled. If Early Depth Stencil Test is not enabled, HW uses NORMAL Input Coverage Masks.
Value	Name	Description														
0h	NONE	Pixel shader does not use input coverage masks.														
1h	NORMAL	Input Coverage masks based on outer conservatism and factors in SAMPLE_MASKs. If Pixel is conservatively covered, all samples are enabled.														
2h	INNER_CONSERVATIVE	Input Coverage masks based on inner conservatism and factors in SAMPLE_MASKs. If Pixel is conservatively fully covered all samples are enabled.														
3h	DEPTH_COVERAGE	Input coverage masks are computed after factoring depth/stencil test results, only if Early Depth Stencil Test is enabled. If Early Depth Stencil Test is not enabled, HW uses NORMAL Input Coverage Masks.														



3DSTATE_PUSH_CONSTANT_ALLOC_DS

3DSTATE_PUSH_CONSTANT_ALLOC_DS		
Source:	RenderCS	
Length Bias:	2	
This command sets up the URB configuration for DS Push Constant Buffer.		
Programming Notes		
Programming Restriction: <ul style="list-style-type: none"> 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_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. The 3DSTATE_CONSTANT_DS must be committed as state prior to shaders generating thread payloads after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_DS. 		
When gather at set shader is disabled, 3DSTATE_CONSTANT command is committed when 3DPRIMITIVE command is parsed. When gather at set shader is enabled, commit point on the 3DSTATE_CONSTANT command is a 3DSTATE_BINDING_TABLE_POINTER command.		
The 3DSTATE_BINDING_TABLE_POINTER_DS must be reprogrammed prior to the next 3DPRIMITIVE command after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_DS if 3DSTATE_CONSTANT_VS command is committed on a binding table pointer due to Gather at Set Shader feature.		
DWord	Bit	Description
0 Programming Notes: Workaround: This command must be followed by a PIPE_CONTROL with CS Stall bit set.,	31:29	Command Type
		Default Value:
	Format:	OpCode
	28:27	Command SubType
		Default Value:
	Format:	OpCode
	26:24	3D Command Opcode
		Default Value:
	Format:	OpCode
	23:16	3D Command Sub Opcode
Default Value:		14h 3DSTATE_PUSH_CONSTANT_ALLOC_DS
Format:	OpCode	
15:8	Reserved	
	Format:	MBZ



3DSTATE_PUSH_CONSTANT_ALLOC_DS										
	<table border="1"> <tr> <td style="text-align: center;">7:0</td> <td>DWord Length</td> </tr> <tr> <td>Default Value:</td> <td>0h Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table>	7:0	DWord Length	Default Value:	0h Excludes DWord (0,1)	Format:	=n Total Length - 2			
7:0	DWord Length									
Default Value:	0h Excludes DWord (0,1)									
Format:	=n Total Length - 2									
1	<table border="1"> <tr> <td style="text-align: center;">31:21</td> <td>Reserved</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	31:21	Reserved	Format:	MBZ					
	31:21	Reserved								
	Format:	MBZ								
	<table border="1"> <tr> <td style="text-align: center;">20:16</td> <td>Constant Buffer Offset</td> </tr> <tr> <td>Format:</td> <td>U5</td> </tr> <tr> <td colspan="2">Specifies the offset of the DS constant buffer into the URB.</td> </tr> <tr> <td style="text-align: center;">Value</td> <td style="text-align: center;">Name</td> </tr> <tr> <td>[0,31]</td> <td>(0KB - 31KB) Increments of 2KB</td> </tr> </table>	20:16	Constant Buffer Offset	Format:	U5	Specifies the offset of the DS constant buffer into the URB.		Value	Name	[0,31]
20:16	Constant Buffer Offset									
Format:	U5									
Specifies the offset of the DS constant buffer into the URB.										
Value	Name									
[0,31]	(0KB - 31KB) Increments of 2KB									
<table border="1"> <tr> <td style="text-align: center;">15:6</td> <td>Reserved</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	15:6	Reserved	Format:	MBZ						
15:6	Reserved									
Format:	MBZ									
<table border="1"> <tr> <td style="text-align: center;">5:0</td> <td>Constant Buffer Size</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> <tr> <td colspan="2">Specifies the size of the DS constant buffer. This value will determine the amount of data the command stream can pre-fetch before the buffer is full. Value of zero is only valid when constants are not enabled for DS.</td> </tr> <tr> <td style="text-align: center;">Value</td> <td style="text-align: center;">Name</td> </tr> <tr> <td>[0,32]</td> <td>(0KB - 32KB) Increments of 2KB</td> </tr> </table>	5:0	Constant Buffer Size	Format:	U6	Specifies the size of the DS constant buffer. This value will determine the amount of data the command stream can pre-fetch before the buffer is full. Value of zero is only valid when constants are not enabled for DS.		Value	Name	[0,32]	(0KB - 32KB) Increments of 2KB
5:0	Constant Buffer Size									
Format:	U6									
Specifies the size of the DS constant buffer. This value will determine the amount of data the command stream can pre-fetch before the buffer is full. Value of zero is only valid when constants are not enabled for DS.										
Value	Name									
[0,32]	(0KB - 32KB) Increments of 2KB									



3DSTATE_PUSH_CONSTANT_ALLOC_GS

3DSTATE_PUSH_CONSTANT_ALLOC_GS		
Source:	RenderCS	
Length Bias:	2	
This command sets up the URB configuration for GS Push Constant Buffer.		
Programming Notes		
<ul style="list-style-type: none"> 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 committed as state prior to shaders generating thread payloads after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_GS. 		
See Push Constant URB Allocation section for more details.		
When gather at set shader is disabled, 3DSTATE_CONSTANT command is committed when 3DPRIMITIVE command is parsed. When gather at set shader is enabled, commit point on the 3DSTATE_CONSTANT command is a 3DSTATE_BINDING_TABLE_POINTER command.		
The 3DSTATE_BINDING_TABLE_POINTER_GS must be reprogrammed prior to the next 3DPRIMITIVE command after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_GS if 3DSTATE_CONSTANT_VS command is committed on a binding table pointer due to Gather at Set Shader feature.		
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: OpCode		
23:16	3D Command Sub Opcode	
	Default Value: 15h 3DSTATE_PUSH_CONSTANT_ALLOC_GS	
Format: OpCode		
15:8	Reserved	
	Format: MBZ	



3DSTATE_PUSH_CONSTANT_ALLOC_GS										
	7:0	<p>DWord Length</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>=n</td> </tr> </table> <p>Total Length - 2</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 60%;">Name</th> <th style="width: 25%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>3DSTATE_PUSH_CONSTANT_ALLOC_GS [Default]</td> <td>Excludes DWord (0,1)</td> </tr> </tbody> </table>	Format:	=n	Value	Name	Description	0h	3DSTATE_PUSH_CONSTANT_ALLOC_GS [Default]	Excludes DWord (0,1)
	Format:	=n								
	Value	Name	Description							
	0h	3DSTATE_PUSH_CONSTANT_ALLOC_GS [Default]	Excludes DWord (0,1)							
1	31:21	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
	Format:	MBZ								
	20:16	<p>Constant Buffer Offset</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Specifies the offset of the GS constant buffer into the URB.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 85%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td>(0KB - 31KB) Increments of 2KB</td> </tr> </tbody> </table>	Format:	U5	Value	Name	[0,31]	(0KB - 31KB) Increments of 2KB		
		Format:	U5							
		Value	Name							
[0,31]	(0KB - 31KB) Increments of 2KB									
<table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ									
15:6	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
5:0	<p>Constant Buffer Size</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Specifies the size of the GS constant buffer. This value will determine the amount of data the command stream can pre-fetch before the buffer is full. Value of zero is only valid when constants are not enabled for GS.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 85%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,32]</td> <td>(0KB - 32KB) Increments of 2KB</td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,32]	(0KB - 32KB) Increments of 2KB			
	Format:	U6								
	Value	Name								
[0,32]	(0KB - 32KB) Increments of 2KB									
<table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ									



3DSTATE_PUSH_CONSTANT_ALLOC_HS

3DSTATE_PUSH_CONSTANT_ALLOC_HS		
Source:	RenderCS	
Length Bias:	2	
This command sets up the URB configuration for HS Push Constant Buffer.		
Programming Notes		
<p>Programming Restriction:</p> <ul style="list-style-type: none"> 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_HS 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_HS must be committed as state prior to shaders generating thread payloads after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_HS. 		
<p>When gather at set shader is disabled, 3DSTATE_CONSTANT command is committed when 3DPRIMITIVE command is parsed.</p> <p>When gather at set shader is enabled, commit point on the 3DSTATE_CONSTANT command is a 3DSTATE_BINDING_TABLE_POINTER command.</p>		
<p>The 3DSTATE_BINDING_TABLE_POINTER_HS must be reprogrammed prior to the next 3DPRIMITIVE command after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_HS if 3DSTATE_CONSTANT_HS command is committed on a binding table pointer due to Gather at Set Shader feature.</p>		
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: OpCode	
	23:16	3D Command Sub Opcode
		Default Value: 13h 3DSTATE_PUSH_CONSTANT_ALLOC_HS
	Format: OpCode	
15:8	Reserved	
	Format: MBZ	



3DSTATE_PUSH_CONSTANT_ALLOC_HS							
	7:0	DWord Length					
	<table border="1"> <tr> <td>Default Value:</td> <td>0h Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table>		Default Value:	0h Excludes DWord (0,1)	Format:	=n Total Length - 2	
Default Value:	0h Excludes DWord (0,1)						
Format:	=n Total Length - 2						
1	31:21	Reserved					
	<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ			
	Format:	MBZ					
	20:16	Constant Buffer Offset					
	<table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Specifies the offset of the HS constant buffer into the URB.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td>(0KB - 31KB) Increments of 2KB</td> </tr> </tbody> </table>		Format:	U5	Value	Name	[0,31]
Format:	U5						
Value	Name						
[0,31]	(0KB - 31KB) Increments of 2KB						
15:6	Reserved						
<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ				
Format:	MBZ						
5:0	Constant Buffer Size						
<table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Specifies the size of the HS constant buffer. This value will determine the amount of data the command stream can pre-fetch before the buffer is full. Value of zero is only valid when constants are not enabled for HS.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,32]</td> <td>(0KB - 32KB) Increments of 2KB</td> </tr> </tbody> </table>		Format:	U6	Value	Name	[0,32]	(0KB - 32KB) Increments of 2KB
Format:	U6						
Value	Name						
[0,32]	(0KB - 32KB) Increments of 2KB						



3DSTATE_PUSH_CONSTANT_ALLOC_PS

3DSTATE_PUSH_CONSTANT_ALLOC_PS		
Source:	RenderCS	
Length Bias:	2	
This command sets up the URB configuration for PS Push Constant Buffer.		
Programming Notes		
<p>Restriction:</p> <ul style="list-style-type: none"> 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 committed as state prior to shaders generating thread payloads after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_PS. 		
<p>When gather at set shader is disabled, 3DSTATE_CONSTANT command is committed when 3DPRIMITIVE command is parsed.</p> <p>When gather at set shader is enabled, commit point on the 3DSTATE_CONSTANT command is a 3DSTATE_BINDING_TABLE_POINTER command.</p>		
<p>The 3DSTATE_BINDING_TABLE_POINTER_PS must be reprogrammed prior to the next 3DPRIMITIVE command after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_PS if 3DSTATE_CONSTANT_VS command is committed on a binding table pointer due to Gather at Set Shader feature.</p>		
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: OpCode	
	23:16	3D Command Sub Opcode
		Default Value: 16h 3DSTATE_PUSH_CONSTANT_ALLOC_PS
	Format: OpCode	
15:8	Reserved	
	Format: MBZ	



3DSTATE_PUSH_CONSTANT_ALLOC_PS							
	7:0	Dword Length					
	<table border="1"> <tr> <td>Default Value:</td> <td>0h Excludes Dword (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table>		Default Value:	0h Excludes Dword (0,1)	Format:	=n Total Length - 2	
Default Value:	0h Excludes Dword (0,1)						
Format:	=n Total Length - 2						
1	31:21	Reserved					
	<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ			
	Format:	MBZ					
	20:16	Constant Buffer Offset					
	<table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Specifies the offset of the PS constant buffer into the URB.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td>(0KB - 31KB) Increments of 2KB</td> </tr> </tbody> </table>		Format:	U5	Value	Name	[0,31]
Format:	U5						
Value	Name						
[0,31]	(0KB - 31KB) Increments of 2KB						
15:6	Reserved						
<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ				
Format:	MBZ						
5:0	Constant Buffer Size						
<table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Specifies the size of the PS constant buffer. This value will determine the amount of data the command stream can pre-fetch before the buffer is full. Value of zero is only valid when constants are not enabled for PS.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,32]</td> <td>(0KB - 32KB) Increments of 2KB</td> </tr> </tbody> </table>		Format:	U6	Value	Name	[0,32]	(0KB - 32KB) Increments of 2KB
Format:	U6						
Value	Name						
[0,32]	(0KB - 32KB) Increments of 2KB						



3DSTATE_PUSH_CONSTANT_ALLOC_VS

3DSTATE_PUSH_CONSTANT_ALLOC_VS			
Source:	RenderCS		
Length Bias:	2		
This command sets up the URB configuration for VS Push Constant Buffer.			
Programming Notes			
Programming Restriction: <ul style="list-style-type: none"> 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 committed as state prior to shaders generating thread payloads after programming the 3DSTATE_PUSH_CONSTANT_ALLOC_VS. 			
When gather at set shader is disabled, 3DSTATE_CONSTANT command is committed when 3DPRIMITIVE command is parsed. When gather at set shader is enabled, commit point on the 3DSTATE_CONSTANT command is a 3DSTATE_BINDING_TABLE_POINTER command.			
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:	OpCode
	23:16	3D Command Sub Opcode	
		Default Value:	12h 3DSTATE_PUSH_CONSTANT_ALLOC_VS
		Format:	OpCode
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	



3DSTATE_PUSH_CONSTANT_ALLOC_VS						
1	31:21	Reserved Format: MBZ				
	20:16	Constant Buffer Offset Format: U5 Specifies the offset of the VS constant buffer into the URB. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,31]</td> <td style="text-align: center;">(0KB - 31KB) Increments of 2KB</td> </tr> </tbody> </table>	Value	Name	[0,31]	(0KB - 31KB) Increments of 2KB
	Value	Name				
	[0,31]	(0KB - 31KB) Increments of 2KB				
	15:6	Reserved Format: MBZ				
	5:0	Constant Buffer Size Format: U6 Specifies the size of the VS constant buffer. This value will determine the amount of data the command stream can pre-fetch before the buffer is full. Value of zero is only valid when constants are not enabled for VS. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,32]</td> <td style="text-align: center;">(0KB - 32KB) Increments of 2KB</td> </tr> </tbody> </table>	Value	Name	[0,32]	(0KB - 32KB) Increments of 2KB
	Value	Name				
	[0,32]	(0KB - 32KB) Increments of 2KB				



3DSTATE_RASTER

3DSTATE_RASTER			
Source:		RenderCS	
Length 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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		50h 3DSTATE_RASTER	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	03h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31:28	Reserved	
		Format:	MBZ
	27	Reserved	
		Format:	MBZ
	26	Viewport Z Far Clip Test Enable	
		Format:	Enable This field is used to control whether the Viewport Z Far extent is considered in VertexClipTest.
25	Reserved		
	Format:	MBZ	
24	Conservative Rasterization Enable		
	Format:	Enable This field when set enables conservative rasterization rules for all primitives except rectangles,	



3DSTATE_RASTER																														
		<p>points and lines. For rectangle, points and lines, setting this bit is ignored by hardware.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> This bit must not be set for primitives with poly-stippling enabled. When this bit is set, sampling mode must be set to "Centre" sampling i.e 3DSTATE_MULTISAMPLE::Pixel Location set to CENTER </td> </tr> </tbody> </table>	Programming Notes		This bit must not be set for primitives with poly-stippling enabled. When this bit is set, sampling mode must be set to "Centre" sampling i.e 3DSTATE_MULTISAMPLE::Pixel Location set to CENTER																									
Programming Notes																														
This bit must not be set for primitives with poly-stippling enabled. When this bit is set, sampling mode must be set to "Centre" sampling i.e 3DSTATE_MULTISAMPLE::Pixel Location set to CENTER																														
23:22	<p>API Mode Software sets this field according to the API's version. These bits are set for DX9 or OGL/DX10.0/DX10.1+/DX11.1 per the following values.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>DX9/OGL</td> </tr> <tr> <td>1h</td> <td>DX10.0</td> </tr> <tr> <td>2h</td> <td>DX10.1+</td> </tr> <tr> <td>3h</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	0h	DX9/OGL	1h	DX10.0	2h	DX10.1+	3h	Reserved																			
Value	Name																													
0h	DX9/OGL																													
1h	DX10.0																													
2h	DX10.1+																													
3h	Reserved																													
21	<p>Front Winding Determines whether a triangle object is considered "front facing" if the screen space vertex positions, when traversed in the order, result in a clockwise (CW) or counter-clockwise (CCW) winding order. Does not apply to points or lines.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Clockwise</td> <td>FRONTWINDING_CW</td> </tr> <tr> <td>1h</td> <td>Counter Clockwise [Default]</td> <td>FRONTWINDING_CCW</td> </tr> </tbody> </table>	Value	Name	Description	0h	Clockwise	FRONTWINDING_CW	1h	Counter Clockwise [Default]	FRONTWINDING_CCW																				
Value	Name	Description																												
0h	Clockwise	FRONTWINDING_CW																												
1h	Counter Clockwise [Default]	FRONTWINDING_CCW																												
20:18	<p>Forced Sample Count Format: U3 Enumerated Type</p> <p>This field specifies how many samples/pixel exist for RT Independent Rasterization</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NUMRASTSAMPLES_0</td> <td>No RT Independent Rasterization</td> </tr> <tr> <td>1h</td> <td>NUMRASTSAMPLES_1</td> <td>1 rast-sample/pixel</td> </tr> <tr> <td>2h</td> <td>NUMRASTSAMPLES_2</td> <td>2 rast-samples/pixel</td> </tr> <tr> <td>3h</td> <td>NUMRASTSAMPLES_4</td> <td>4 rast-samples/pixel</td> </tr> <tr> <td>4h</td> <td>NUMRASTSAMPLES_8</td> <td>8 rast-samples/pixel</td> </tr> <tr> <td>5h</td> <td>NUMRASTSAMPLES_16</td> <td>16 rast-samples/pixel</td> </tr> <tr> <td>6h-7h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> When 3DSTATE_MULTISAMPLE::Number of Multisamples != NUMSAMPLES_1, this field must be either NUMRASTSAMPLES_0 or NUMRASTSAMPLES_1. When 3DSTATE_MULTISAMPLE::Number of Multisamples == NUMSAMPLES_1, this field must not be NUMRASTSAMPLES_1. </td> </tr> </tbody> </table>	Value	Name	Description	0h	NUMRASTSAMPLES_0	No RT Independent Rasterization	1h	NUMRASTSAMPLES_1	1 rast-sample/pixel	2h	NUMRASTSAMPLES_2	2 rast-samples/pixel	3h	NUMRASTSAMPLES_4	4 rast-samples/pixel	4h	NUMRASTSAMPLES_8	8 rast-samples/pixel	5h	NUMRASTSAMPLES_16	16 rast-samples/pixel	6h-7h	Reserved		Programming Notes		When 3DSTATE_MULTISAMPLE::Number of Multisamples != NUMSAMPLES_1, this field must be either NUMRASTSAMPLES_0 or NUMRASTSAMPLES_1. When 3DSTATE_MULTISAMPLE::Number of Multisamples == NUMSAMPLES_1, this field must not be NUMRASTSAMPLES_1.		
Value	Name	Description																												
0h	NUMRASTSAMPLES_0	No RT Independent Rasterization																												
1h	NUMRASTSAMPLES_1	1 rast-sample/pixel																												
2h	NUMRASTSAMPLES_2	2 rast-samples/pixel																												
3h	NUMRASTSAMPLES_4	4 rast-samples/pixel																												
4h	NUMRASTSAMPLES_8	8 rast-samples/pixel																												
5h	NUMRASTSAMPLES_16	16 rast-samples/pixel																												
6h-7h	Reserved																													
Programming Notes																														
When 3DSTATE_MULTISAMPLE::Number of Multisamples != NUMSAMPLES_1, this field must be either NUMRASTSAMPLES_0 or NUMRASTSAMPLES_1. When 3DSTATE_MULTISAMPLE::Number of Multisamples == NUMSAMPLES_1, this field must not be NUMRASTSAMPLES_1.																														



3DSTATE_RASTER																	
17:16	<p>Cull Mode Controls removal (culling) of triangle objects based on orientation. The cull mode only applies to triangle objects and does not apply to lines, points or rectangles.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>CULLMODE_BOTH</td> <td>All triangles are discarded (i.e., no triangle objects are drawn)</td> </tr> <tr> <td>1h</td> <td>CULLMODE_NONE [Default]</td> <td>No triangles are discarded due to orientation</td> </tr> <tr> <td>2h</td> <td>CULLMODE_FRONT</td> <td>Triangles with a front-facing orientation are discarded</td> </tr> <tr> <td>3h</td> <td>CULLMODE_BACK</td> <td>Triangles with a back-facing orientation are discarded</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Orientation determination is based on the setting of the Front Winding state.</p>		Value	Name	Description	0h	CULLMODE_BOTH	All triangles are discarded (i.e., no triangle objects are drawn)	1h	CULLMODE_NONE [Default]	No triangles are discarded due to orientation	2h	CULLMODE_FRONT	Triangles with a front-facing orientation are discarded	3h	CULLMODE_BACK	Triangles with a back-facing orientation are discarded
Value	Name	Description															
0h	CULLMODE_BOTH	All triangles are discarded (i.e., no triangle objects are drawn)															
1h	CULLMODE_NONE [Default]	No triangles are discarded due to orientation															
2h	CULLMODE_FRONT	Triangles with a front-facing orientation are discarded															
3h	CULLMODE_BACK	Triangles with a back-facing orientation are discarded															
15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ													
Format:	MBZ																
14	<p>Force Multisampling This field provides a work around override for the computation of SF_INT::Multisample Rasterization Mode and WM_INT::Multisample Rasterization Mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Normal</td> <td>Multisampling mode is computed by HW according to formula for signal SF_INT::Multisample Rasterization Mode and WM_INT::Multisample Rasterization Mode in vol2a.11 3D Pipeline Windower > Windower Pipelined State > 3DSTATE_WM > 3DSTATE_WM.</td> </tr> <tr> <td>1h</td> <td>Force</td> <td>Forces the DX Multisampling mode to be used directly</td> </tr> </tbody> </table>		Value	Name	Description	0h	Normal	Multisampling mode is computed by HW according to formula for signal SF_INT::Multisample Rasterization Mode and WM_INT::Multisample Rasterization Mode in vol2a.11 3D Pipeline Windower > Windower Pipelined State > 3DSTATE_WM > 3DSTATE_WM.	1h	Force	Forces the DX Multisampling mode to be used directly						
Value	Name	Description															
0h	Normal	Multisampling mode is computed by HW according to formula for signal SF_INT::Multisample Rasterization Mode and WM_INT::Multisample Rasterization Mode in vol2a.11 3D Pipeline Windower > Windower Pipelined State > 3DSTATE_WM > 3DSTATE_WM.															
1h	Force	Forces the DX Multisampling mode to be used directly															
13	<p>Smooth Point Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Software sets this according to API. When OGL and smooth point rasterization is required, this bit must be set. HW ignores this bit for primitives other than points.</p>		Format:	Enable													
Format:	Enable																
12	<p>DX Multisample Rasterization Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Software sets this according to the API's multisample enable</p> <p style="text-align: center;">Programming Notes</p> <p>This state only effects how the SF_INT/WM_INT::Multisample Rasterization Mode are set depending on some other states. This state mainly modifies the how the line rendering is done by setting SF_INT/WM_INT::Multisample Rasterization Mode to either OFF* or ON* . Please refer to table under SF_INT::Multisample Rasterization Mode.</p>		Format:	Enable													
Format:	Enable																
11:10	<p>DX Multisample Rasterization Mode</p> <table border="1"> <tr> <td>Format:</td> <td>U2 enumerated type</td> </tr> </table> <p>This field determines whether multisample rasterization is turned on/off, and how the pixel</p>		Format:	U2 enumerated type													
Format:	U2 enumerated type																



3DSTATE_RASTER		
	sample point(s) are defined. Software sets this according to the API's multisample state setting (if any)	
	Value	Name
	0h	MSRASTMODE_OFF_PIXEL
	1h	MSRASTMODE_OFF_PATTERN
	2h	MSRASTMODE_ON_PIXEL
	3h	MSRASTMODE_ON_PATTERN
	Programming Notes	
	This field is used to directly set the SF_INT/WM_INT::Multisample Rasterization Mode when DX Multisample Rasterization Enable is set. Please refer to equation of SF_INT::Multisample Rasterization Mode.	
9	Global Depth Offset Enable Solid	
	Format:	Enable
	Enables computation and application of Global Depth Offset for SOLID objects.	
8	Global Depth Offset Enable Wireframe	
	Format:	Enable
	Enables computation and application of Global Depth Offset when triangles are rendered in WIREFRAME mode.	
7	Global Depth Offset Enable Point	
	Format:	Enable
	Enables computation and application of Global Depth Offset when triangles are rendered in POINT mode.	
6:5	Front Face Fill Mode	
	Format:	U2 enumerated type
	This state controls how front-facing triangle and rectangle objects are rendered.	
	Value	Name
	Value	Description
	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 (RECTLIST) 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 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).
	3h	Reserved



3DSTATE_RASTER		
4:3	Back Face Fill Mode	
	Format: U2 enumerated type	
	This state controls how 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 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).
3h	Reserved	
2	Antialiasing Enable	
	Format: Enable	
	This field enables "alpha-based" line antialiasing.	
	Programming Notes	
		This field must be disabled if any of the render targets have integer (UINT or SINT) surface format.
1	Scissor Rectangle Enable	
	Format: Enable	
		Enables operation of Scissor Rectangle.
0	Viewport Z Near Clip Test Enable	
	Format: Enable	
		This field is used to control whether the Viewport Z Near extent is considered in VertexClipTest.
2	31:0	Global Depth Offset Constant
		Format: IEEE_Float
		Specifies the constant term in the Global Depth Offset function.
3	31:0	Global Depth Offset Scale
		Format: IEEE_Float
		Specifies the scale term used in the Global Depth Offset function.
4	31:0	Global Depth Offset Clamp
		Format: IEEE_Float
		Specifies the clamp term used in the Global Depth Offset function.



3DSTATE_RS_CONSTANT_POINTER

3DSTATE_RS_CONSTANT_POINTER			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_RS_CONSTANT_POINTER command allows the driver to program an indirect address to either store or load the global section of the RS constant buffers.			
Programming Notes			
This command will be treated as a NOOP if the Dx9 Constant Buffer Pool is not enabled by the 3DSTATE_DX9_CONST_POOL command.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	54h 3DSTATE_RS_CONSTANT_POINTER	
	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
	30:28	Shader Select	
		Format:	U3
		Specifies which shaders global register to load or store.	
		Value	Name
0h		VS	Select VS Global Constants
1h-3h	Reserved		
4h	PS	Select PS Global Constants	



3DSTATE_RS_CONSTANT_POINTER											
	5h-7h	Reserved									
	27:13	Reserved									
		Format: MBZ									
	12	<p>Operation Load or Store</p> <p>When this bit is set, the resource streamer will load the global constants to the pointer to the global buffer in the hardware based on the Shader Select bit.</p> <p>When this bit is clear, the resource streamer will store the global constants to the pointer from the global buffer in the hardware based on the Shader Select bit.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Store</td> <td>Store buffer to memory</td> </tr> <tr> <td>1h</td> <td>Load</td> <td>Load buffer from memory</td> </tr> </tbody> </table>	Value	Name	Description	0h	Store	Store buffer to memory	1h	Load	Load buffer from memory
	Value	Name	Description								
0h	Store	Store buffer to memory									
1h	Load	Load buffer from memory									
	Format: MBZ										
2	11:0	Reserved									
		Format: MBZ									
	31:6	<p>Global Constant Buffer Address</p> <p>Format: GraphicsAddress[31:6]</p> <p>Specifies the Cache Line Aligned Graphics address of location in memory of where to load or store a copy of the global constant buffer.</p>									
	5:0	Reserved									
		Format: MBZ									
3	31:0	<p>Global Constant Buffer Address High</p> <p>Format: GraphicsAddress[63:32]</p> <p>Specifies the Upper portion of the graphics address location in memory of where to load or store a copy of the global constant buffer. Upper 16 bits MBZ.</p>									



3DSTATE_SAMPLE_MASK

3DSTATE_SAMPLE_MASK			
Source:		RenderCS	
Length 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:	0h 3DSTATE_PIPELINED
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	18h 3DSTATE_SAMPLE_MASK	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	Dword Length		
	Default Value:	0h Excludes Dword (0,1)	
	Format:	=n Total Length - 2	
1	31:16	Reserved	
		Format:	MBZ
	15:0	Sample Mask	
		Format:	16 bit mask Right-justified bitmask (Bit 0 = Sample0). Number of bits that are used is determined by Num Multisamples (3DSTATE_MULTISAMPLE)
		<p>A per-multisample-position mask state variable that is immediately and unconditionally ANDed with the sample coverage mask as part of the rasterization process. This mask is applied prior to centroid selection. This mask must be ignored for centroid selection when RTIR is enabled i.e. Forced_Sample_Count > 0.</p>	
		<p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> • If Number of Multisamples is NUMSAMPLES_1, bits 15:1 of this field will be zeroed by HW. • If Number of Multisamples is NUMSAMPLES_2, bits 15:2 of this field will be zeroed by HW. • If Number of Multisamples is NUMSAMPLES_4, bits 15:4 of this field will be zeroed by 	



3DSTATE_SAMPLE_MASK

HW.

- If **Number of Multisamples** is NUMSAMPLES_8, bits 15:8 of this field will be zeroed by HW.

When pixel shader writes to UAV but does not have actual render target write (i.e. no RT is bound to pixel shader, eventhough, RT write message is sent for EOT), appropriate SAMPLE_MASK must be all set depending on Number of Multisamples.



3DSTATE_SAMPLE_PATTERN

3DSTATE_SAMPLE_PATTERN			
Source:	RenderCS		
Length Bias:	2		
<p>The 3DSTATE_SAMPLE_PATTERN command is used to specify the sample offsets for all multisample sample modes.</p> <p>The set of offset used will be selected based on the multisample mode. This is non-pipelined state.</p>			
Programming Notes			
<p>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, or 15 for 16X) must have monotonically increasing distance from the pixel center. This is required to get the correct centroid computation in the device.</p>			
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:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	1Ch 3DSTATE_SAMPLE_PATTERN	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	7 Excludes Dword (0,1)	
	Format:	=n Total Length - 2	
1	31:28	16x Sample3 X Offset	
		Format:	U0.4
		Subpixel X offset of Sample 3 relative to the UL pixel origin for 16x mode.	
		Range: [0,0.9375]	



3DSTATE_SAMPLE_PATTERN		
27:24	16x Sample3 Y Offset	
	Format: U0.4	
	Subpixel Y offset of Sample 3 relative to the UL pixel origin for 16x mode.	
	Range: [0,0.9375]	
	16x Sample2 X Offset	
	Format: U0.4	
23:20	Subpixel X offset of Sample 2 relative to the UL pixel origin for 16x mode.	
	Range: [0,0.9375]	
	16x Sample2 Y Offset	
19:16	Format: U0.4	
	Subpixel Y offset of Sample 2 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_4, _8 or _16.	
	Range: [0,0.9375]	
15:12	16x Sample1 X Offset	
	Format: U0.4	
	Subpixel X offset of Sample 1 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_2, _4, _8 or _16.	
11:8	Range: [0,0.9375]	
	16x Sample1 Y Offset	
	Format: U0.4	
7:4	Subpixel Y offset of Sample 1 relative to the UL pixel origin for 16x mode.	
	Range: [0,0.9375]	
	16x Sample0 X Offset	
3:0	Format: U0.4	
	Subpixel X offset of Sample 0 relative to the UL pixel origin for 16x mode.	
	Range: [0,0.9375]	
3:0	16x Sample0 Y Offset	
	Format: U0.4	
	Subpixel Y offset of Sample 0 relative to the UL pixel origin for 16x mode.	
3:0	Range: [0,0.9375]	



3DSTATE_SAMPLE_PATTERN								
2	31:28	<p>16x Sample7 X Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel X offset of Sample 7 relative to the UL pixel origin for 16x mode.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel X offset of Sample 7 relative to the UL pixel origin for 16x mode.		Range: [0,0.9375]	
	Format:	U0.4						
	Subpixel X offset of Sample 7 relative to the UL pixel origin for 16x mode.							
	Range: [0,0.9375]							
	27:24	<p>16x Sample7 Y Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel Y offset of Sample 7 relative to the UL pixel origin for 16x mode.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel Y offset of Sample 7 relative to the UL pixel origin for 16x mode.		Range: [0,0.9375]	
	Format:	U0.4						
	Subpixel Y offset of Sample 7 relative to the UL pixel origin for 16x mode.							
Range: [0,0.9375]								
23:20	<p>16x Sample6 X Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel X offset of Sample 6 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_8 or _16.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel X offset of Sample 6 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_8 or _16.		Range: [0,0.9375]		
Format:	U0.4							
Subpixel X offset of Sample 6 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_8 or _16.								
Range: [0,0.9375]								
19:16	<p>16x Sample6 Y Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel Y offset of Sample 6 relative to the UL pixel origin for 16x mode.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel Y offset of Sample 6 relative to the UL pixel origin for 16x mode.		Range: [0,0.9375]		
Format:	U0.4							
Subpixel Y offset of Sample 6 relative to the UL pixel origin for 16x mode.								
Range: [0,0.9375]								
15:12	<p>16x Sample5 X Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel X offset of Sample 5 relative to the UL pixel origin for 16x mode.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel X offset of Sample 5 relative to the UL pixel origin for 16x mode.		Range: [0,0.9375]		
Format:	U0.4							
Subpixel X offset of Sample 5 relative to the UL pixel origin for 16x mode.								
Range: [0,0.9375]								
11:8	<p>16x Sample5 Y Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel Y offset of Sample 5 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_8 or _16.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel Y offset of Sample 5 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_8 or _16.		Range: [0,0.9375]		
Format:	U0.4							
Subpixel Y offset of Sample 5 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_8 or _16.								
Range: [0,0.9375]								
7:4	<p>16x Sample4 X Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel X offset of Sample 4 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_8 or _16.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel X offset of Sample 4 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_8 or _16.		Range: [0,0.9375]		
Format:	U0.4							
Subpixel X offset of Sample 4 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_8 or _16.								
Range: [0,0.9375]								



3DSTATE_SAMPLE_PATTERN								
3	3:0	16x Sample4 Y Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel Y offset of Sample 4 relative to the UL pixel origin for 16x mode.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel Y offset of Sample 4 relative to the UL pixel origin for 16x mode.		Range: [0,0.9375]	
	Format:	U0.4						
	Subpixel Y offset of Sample 4 relative to the UL pixel origin for 16x mode.							
	Range: [0,0.9375]							
	31:28	16x Sample11 X Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel X offset of Sample 11 relative to the UL pixel origin for 16x mode.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel X offset of Sample 11 relative to the UL pixel origin for 16x mode.		Range: [0,0.9375]	
	Format:	U0.4						
	Subpixel X offset of Sample 11 relative to the UL pixel origin for 16x mode.							
	Range: [0,0.9375]							
27:24	16x Sample11 Y Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel Y offset of Sample 11 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_16.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel Y offset of Sample 11 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_16.		Range: [0,0.9375]		
Format:	U0.4							
Subpixel Y offset of Sample 11 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_16.								
Range: [0,0.9375]								
23:20	16x Sample10 X Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel X offset of Sample 10 relative to the UL pixel origin for 16x mode.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel X offset of Sample 10 relative to the UL pixel origin for 16x mode.		Range: [0,0.9375]		
Format:	U0.4							
Subpixel X offset of Sample 10 relative to the UL pixel origin for 16x mode.								
Range: [0,0.9375]								
19:16	16x Sample10 Y Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel Y offset of Sample 10 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_16</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel Y offset of Sample 10 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_16		Range: [0,0.9375]		
Format:	U0.4							
Subpixel Y offset of Sample 10 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_16								
Range: [0,0.9375]								
15:12	16x Sample9 X Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel X offset of Sample 9 relative to the UL pixel origin for 16x mode.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel X offset of Sample 9 relative to the UL pixel origin for 16x mode.		Range: [0,0.9375]		
Format:	U0.4							
Subpixel X offset of Sample 9 relative to the UL pixel origin for 16x mode.								
Range: [0,0.9375]								
11:8	16x Sample9 Y Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.4</td> </tr> <tr> <td colspan="2">Subpixel Y offset of Sample 9 relative to the UL pixel origin for 16x mode.</td> </tr> <tr> <td colspan="2">Range: [0,0.9375]</td> </tr> </table>	Format:	U0.4	Subpixel Y offset of Sample 9 relative to the UL pixel origin for 16x mode.		Range: [0,0.9375]		
Format:	U0.4							
Subpixel Y offset of Sample 9 relative to the UL pixel origin for 16x mode.								
Range: [0,0.9375]								



3DSTATE_SAMPLE_PATTERN		
	7:4	16x Sample8 X Offset Format: U0.4 Subpixel X offset of Sample 8 relative to the UL pixel origin for 16x mode. Range: [0,0.9375]
	3:0	16x Sample8 Y Offset Format: U0.4 Subpixel Y offset of Sample 8 relative to the UL pixel origin for 16x mode. Range: [0,0.9375]
4	31:28	16x Sample15 X Offset Format: U0.4 Subpixel X offset of Sample 15 relative to the UL pixel origin for 16x mode. Range: [0,0.9375]
	27:24	16x Sample15 Y Offset Format: U0.4 Subpixel Y offset of Sample 15 relative to the UL pixel origin for 16x mode. Range: [0,0.9375]
	23:20	16x Sample14 X Offset Format: U0.4 Subpixel X offset of Sample 14 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_16. Range: [0,0.9375]
	19:16	16x Sample14 Y Offset Format: U0.4 Subpixel Y offset of Sample 14 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_16 Range: [0,0.9375]
	15:12	16x Sample13 X Offset Format: U0.4 Subpixel X offset of Sample 13 relative to the UL pixel origin for 16x mode. Range: [0,0.9375]



3DSTATE_SAMPLE_PATTERN					
	11:8	16x Sample13 Y Offset	Format: U0.4	Subpixel Y offset of Sample 13 relative to the UL pixel origin for 16x mode. Range: [0,0.9375]	
	7:4	16x Sample12 X Offset	Format: U0.4	Subpixel X offset of Sample 12 relative to the UL pixel origin. Valid only when NUMRASTSAMPLES_16. Range: [0,0.9375]	
	3:0	16x Sample12 Y Offset	Format: U0.4	Subpixel Y offset of Sample 12 relative to the UL pixel origin for 16x mode. Range: [0,0.9375]	
	5	31:28	8x Sample7 X Offset	Format: U0.4	Subpixel X offset of Sample 7 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
		27:24	8x Sample7 Y Offset	Format: U0.4	Subpixel Y offset of Sample 7 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
		23:20	8x Sample6 X Offset	Format: U0.4	Subpixel X offset of Sample 6 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
		19:16	8x Sample6 Y Offset	Format: U0.4	Subpixel Y offset of Sample 6 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]



3DSTATE_SAMPLE_PATTERN				
6	15:12	8x Sample5 X Offset	Format: U0.4	Subpixel X offset of Sample 5 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	11:8	8x Sample5 Y Offset	Format: U0.4	Subpixel Y offset of Sample 5 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	7:4	8x Sample4 X Offset	Format: U0.4	Subpixel X offset of Sample 4 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	3:0	8x Sample4 Y Offset	Format: U0.4	Subpixel Y offset of Sample 4 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	31:28	8x Sample3 X Offset	Format: U0.4	Subpixel X offset of Sample 3 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	27:24	8x Sample3 Y Offset	Format: U0.4	Subpixel Y offset of Sample 3 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	23:20	8x Sample2 X Offset	Format: U0.4	Subpixel X offset of Sample 2 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]



3DSTATE_SAMPLE_PATTERN		
	19:16	8x Sample2 Y Offset Format: U0.4 Subpixel Y offset of Sample 2 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	15:12	8x Sample1 X Offset Format: U0.4 Subpixel X offset of Sample 1 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	11:8	8x Sample1 Y Offset Format: U0.4 Subpixel Y offset of Sample 1 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	7:4	8x Sample0 X Offset Format: U0.4 Subpixel X offset of Sample 0 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
	3:0	8x Sample0 Y Offset Format: U0.4 Subpixel Y offset of Sample 0 relative to the UL pixel origin for 8x mode. Range: [0,0.9375]
7	31:28	4x Sample3 X Offset Format: U0.4 Subpixel X offset of Sample 3 relative to the UL pixel origin for 4x mode. Range: [0,0.9375]
	27:24	4x Sample3 Y Offset Format: U0.4 Subpixel Y offset of Sample 3 relative to the UL pixel origin for 4x mode. Range: [0,0.9375]



3DSTATE_SAMPLE_PATTERN		
	23:20	4x Sample2 X Offset Format: U0.4 Subpixel X offset of Sample 2 relative to the UL pixel origin for 4x mode. Range: [0,0.9375]
	19:16	4x Sample2 Y Offset Format: U0.4 Subpixel Y offset of Sample 2 relative to the UL pixel origin for 4x mode. Range: [0,0.9375]
	15:12	4x Sample1 X Offset Format: U0.4 Subpixel X offset of Sample 1 relative to the UL pixel origin for 4x mode. Range: [0,0.9375]
	11:8	4x Sample1 Y Offset Format: U0.4 Subpixel Y offset of Sample 1 relative to the UL pixel origin for 4x mode. Range: [0,0.9375]
	7:4	4x Sample0 X Offset Format: U0.4 Subpixel X offset of Sample 0 relative to the UL pixel origin for 4x mode. Range: [0,0.9375]
	3:0	4x Sample0 Y Offset Format: U0.4 Subpixel Y offset of Sample 0 relative to the UL pixel origin for 4x mode. Range: [0,0.9375]
8	31:24	Reserved Format: MBZ
	23:20	1x Sample0 X Offset Format: U0.4 Subpixel X offset of Sample 0 relative to the UL pixel origin for 1x mode. Range: [0,0.9375]



3DSTATE_SAMPLE_PATTERN	
19:16	1x Sample0 Y Offset
	Format: U0.4
	Subpixel Y offset of Sample 0 relative to the UL pixel origin for 1x mode.
	Range: [0,0.9375]
15:12	2x Sample1 X Offset
	Format: U0.4
	Subpixel X offset of Sample 1 relative to the UL pixel origin for 2x mode.
	Range: [0,0.9375]
11:8	2x Sample1 Y Offset
	Format: U0.4
	Subpixel Y offset of Sample 1 relative to the UL pixel origin for 2x mode.
	Range: [0,0.9375]
7:4	2x Sample0 X Offset
	Format: U0.4
	Subpixel X offset of Sample 0 relative to the UL pixel origin for 2x mode.
	Range: [0,0.9375]
3:0	2x Sample0 Y Offset
	Format: U0.4
	Subpixel Y offset of Sample 0 relative to the UL pixel origin for 2x mode.
	Range: [0,0.9375]



3DSTATE_SAMPLER_PALETTE_LOAD0

3DSTATE_SAMPLER_PALETTE_LOAD0			
Source:	RenderCS		
Length Bias:	2		
Description			
<p>The 3DSTATE_SAMPLER_PALETTE_LOAD0 instruction is used to load 32-bit values into the first texture palette. The texture palette is used whenever a texture with a paletted format (containing "Px [palette0]") is referenced by the sampler.</p> <p>This instruction is used to load all or a subset of the 256 entries of the first palette. Partial loads always start from the first (index 0) entry.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	Opcode
	28:27	Command SubType	
		Default Value:	3h GFXPIPE_3D
	26:24	3D Command Opcode	
		Default Value:	1h 3DSTATE_NONPIPELINED
23:16	3D Command Sub Opcode		
	Default Value:	02h 3DSTATE_SAMPLER_PALETTE_LOAD0	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Format:	=n	
	Total Length = 1 + entryCount - 2		
	Value	Name	Description
[0,255]	Range	1-256 Entries	
1..n	31:0	Entry	
		Format:	PALETTE_ENTRY



3DSTATE_SAMPLER_PALETTE_LOAD1

3DSTATE_SAMPLER_PALETTE_LOAD1			
Source:	RenderCS		
Length Bias:	2		
<p>The 3DSTATE_SAMPLER_PALETTE_LOAD1 instruction is used to load 32-bit values into the second texture palette. The second texture palette is used whenever a texture with a paletted format (containing "Px...[palette1]") is referenced by the sampler. This instruction is used to load all or a subset of the 256 entries of the second palette. Partial loads always start from the first (index 0) entry.</p>			
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:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		0Ch 3DSTATE_SAMPLER_PALETTE_LOAD1	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1..n	31:24	Palette Alpha[0:N-1]	
	Format:	U8 Alpha channel loaded into the Nth entry of the texture color palette.	
1..n	23:16	Palette Red[0:N-1]	
	Format:	U8 Alpha channel loaded into the Nth entry of the texture color palette.	



3DSTATE_SAMPLER_PALETTE_LOAD1				
	15:8	Palette Green[0:N-1] <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U8</td> </tr> </table> Alpha channel loaded into the Nth entry of the texture color palette.	Format:	U8
	Format:	U8		
7:0	Palette Blue[0:N-1] <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U8</td> </tr> </table> Alpha channel loaded into the Nth entry of the texture color palette.	Format:	U8	
Format:	U8			



3DSTATE_SAMPLER_STATE_POINTERS_DS

3DSTATE_SAMPLER_STATE_POINTERS_DS		
Source:	RenderCS	
Length Bias:	2	
The 3DSTATE_SAMPLER_STATE_POINTERS_DS command is used to define the location of DS SAMPLER_STATE table. Only some of the fixed functions utilize sampler state tables.		
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: 0h 3DSTATE_PIPELINED		
Format: OpCode		
23:16	3D Command Sub Opcode	
	Default Value: 2Dh 3DSTATE_SAMPLER_STATE_POINTERS_DS	
	Format: OpCode	
15:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Default Value: 0h DWORD_COUNT_n	
	Format: =n	
1	31:5	Pointer to DS Sampler State
		Format: DynamicStateOffset[31:5]SAMPLER_STATE*16 Specifies the 32-byte aligned address offset of the DS function's SAMPLER_STATE table. This offset is relative to the Dynamic State Base Address.
	4:0	Reserved
		Format: MBZ



3DSTATE_SAMPLER_STATE_POINTERS_GS

3DSTATE_SAMPLER_STATE_POINTERS_GS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_SAMPLER_STATE_POINTERS_GS command is used to define the location of GS SAMPLER_STATE table. Only some of the fixed functions utilize sampler state tables.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	2Eh 3DSTATE_SAMPLER_STATE_POINTERS_GS	
	Format:	OpCode	
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 aligned address offset of the GS function's SAMPLER_STATE table. This offset is relative to the Dynamic State Base Address.		
4:0	Reserved		
	Format:	MBZ	



3DSTATE_SAMPLER_STATE_POINTERS_HS

3DSTATE_SAMPLER_STATE_POINTERS_HS		
Source:	RenderCS	
Length Bias:	2	
The 3DSTATE_SAMPLER_STATE_POINTERS_HS command is used to define the location of HS SAMPLER_STATE table. Only some of the fixed functions utilize sampler state tables.		
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: 0h 3DSTATE_PIPELINED		
Format: OpCode		
23:16	3D Command Sub Opcode	
	Default Value: 2Ch 3DSTATE_SAMPLER_STATE_POINTERS_HS	
	Format: OpCode	
15:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Default Value: 0h DWORD_COUNT_n	
	Format: =n	
1	31:5	Pointer to HS Sampler State
		Format: DynamicStateOffset[31:5]SAMPLER_STATE*16 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_SAMPLER_STATE_POINTERS_PS

3DSTATE_SAMPLER_STATE_POINTERS_PS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_SAMPLER_STATE_POINTERS_PS command is used to define the location of PS SAMPLER_STATE table. Only some of the fixed functions utilize sampler state tables.			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		2Fh 3DSTATE_SAMPLER_STATE_POINTERS_PS	
Format:		OpCode	
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: DynamicStateOffset[31:5]SAMPLER_STATE*16	
	Specifies the 32-byte aligned address offset of the PS function's SAMPLER_STATE table. This offset is relative to the Dynamic State Base Address.		
4:0	Reserved		
Format:		MBZ	



3DSTATE_SAMPLER_STATE_POINTERS_VS

3DSTATE_SAMPLER_STATE_POINTERS_VS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_SAMPLER_STATE_POINTERS_VS command is used to define the location of VS SAMPLER_STATE table. Only some of the fixed functions utilize sampler state tables.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	2Bh 3DSTATE_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	
		Format: DynamicStateOffset[31:5]SAMPLER_STATE*16 Specifies the 32-byte aligned address offset of the VS function's SAMPLER_STATE table. This offset is relative to the Dynamic State Base Address.	
1	4:0	Reserved	
		Format: MBZ	



3DSTATE_SBE

3DSTATE_SBE			
Source:		RenderCS	
Length 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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		1Fh 3DSTATE_SBE	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	04h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31:30	Reserved	
		Format:	MBZ
	29	Force Vertex URB Entry Read Length	
		Format:	Enable
	This field provides a work around override for the computation of SBE_INT::Vertex URB Entry Read Length. If asserted, 3DSTATE_SBE::Vertex URB Entry Read Length is be used directly. Otherwise, SBE_INT::Vertex URB Entry Read Length is computed normally.		
28	Force Vertex URB Entry Read Offset		
	Format:	Enable	
This field provides a work around override for the computation of SBE_INT::Vertex URB Entry Read Offset. If asserted, 3DSTATE_SBE::Vertex URB Entry Read Offset is be used directly. Otherwise, SBE_INT::Vertex URB Entry Read Offset is computed normally.			



3DSTATE_SBE		
27:22	Number of SF Output Attributes	
	Format:	U6 count of attributes
Specifies the number of vertex attributes passed from the SF stage to the WM stage (does not include Position).		
Value		Name
[0,32]		
21	Attribute Swizzle Enable	
	Format:	Enable
Enables the SF to perform swizzling on (up to the first 16) vertex attributes. If DISABLED, all vertex attributes are passed through.		
20	Point Sprite Texture Coordinate Origin	
	This state controls how Point Sprite Texture Coordinates are generated (when enabled on a per-attribute basis by Point Sprite Texture Coordinate Enable).	
	Value	Name
	Description	
0h	UPPERLEFT	Top Left = (0,0,0,1)Bottom Left = (0,1,0,1)Bottom Right = (1,1,0,1)
1h	LOWERLEFT	Top Left = (0,1,0,1)Bottom Left = (0,0,0,1)Bottom Right = (1,0,0,1)
19	Primitive ID Override Component W	
	Format:	Enable
If set, the W component of output attribute selected by Primitive ID Override Attribute Select is overridden with the Primitive ID.		
18	Primitive ID Override Component Z	
	Format:	Enable
If set, the Z component of output attribute selected by Primitive ID Override Attribute Select is overridden with the Primitive ID.		
17	Primitive ID Override Component Y	
	Format:	Enable
If set, the Y component of output attribute selected by Primitive ID Override Attribute Select is overridden with the Primitive ID.		
16	Primitive ID Override Component X	
	Format:	Enable
If set, the X component of output attribute selected by Primitive ID Override Attribute Select is overridden with the Primitive ID.		
15:11	Vertex URB Entry Read Length	
	Format:	U5
Specifies the amount of URB data read for each Vertex URB entry, in 256-bit register increments.		



		3DSTATE_SBE	
		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)	
	10:5	Vertex URB Entry Read Offset Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB.	
	4:0	Primitive ID Override Attribute Select Specifies which attribute is overridden w/ the Primitive ID	
		Programming Notes	
		Set all Primitive ID Override Component Select X/Y/Z/W to 0 to indicate there is no Primitive ID override.	
2	31:0	Point Sprite Texture Coordinate Enable	
		Format:	Enable[32]
		When processing point primitives, the attributes from the incoming point vertex are typically copied to the point object corner vertices. However, if a bit is set in this field, the corresponding Attribute is selected as a Point Sprite Texture Coordinate, in which case each corner vertex is assigned a pre-defined texture coordinate as defined by the Point Sprite Texture Coordinate Origin state bit. Bit 0 corresponds to output Attribute 0.	
3	31:0	Constant Interpolation Enable	
		Format:	Enable[32]
		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.	
4	31:30	Attribute 15 Active Component Format	
		Format:	Attribute_Component_Format
		This state indicates which components of Attribute 15 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.	
	29:28	Attribute 14 Active Component Format	
		Format:	Attribute_Component_Format
		This state indicates which components of Attribute 14 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is	



3DSTATE_SBE

		UNDEFINED if kernel uses attribute vertex delta for any disabled component.		
27:26	Attribute 13 Active Component Format	<table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 13 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format			
25:24	Attribute 12 Active Component Format	<table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 12 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format			
23:22	Attribute 11 Active Component Format	<table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 11 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format			
21:20	Attribute 10 Active Component Format	<table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 10 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format			
19:18	Attribute 9 Active Component Format	<table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 9 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format			
17:16	Attribute 8 Active Component Format	<table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 8 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format			
15:14	Attribute 7 Active Component Format	<table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 7 are being used by the pixel shader Kernel.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format			



3DSTATE_SBE			
	SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.		
13:12	<p>Attribute 6 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 6 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
11:10	<p>Attribute 5 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 5 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
9:8	<p>Attribute 4 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 4 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
7:6	<p>Attribute 3 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 3 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
5:4	<p>Attribute 2 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 2 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
3:2	<p>Attribute 1 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 1 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
1:0	<p>Attribute 0 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		



3DSTATE_SBE		
		This state indicates which components of Attribute 0 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
5	31:30	Attribute 31 Active Component Format Format: Attribute_Component_Format This state indicates which components of Attribute 31 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	29:28	Attribute 30 Active Component Format Format: Attribute_Component_Format This state indicates which components of Attribute 30 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	27:26	Attribute 29 Active Component Format Format: Attribute_Component_Format This state indicates which components of Attribute 29 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	25:24	Attribute 28 Active Component Format Format: Attribute_Component_Format This state indicates which components of Attribute 28 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	23:22	Attribute 27 Active Component Format Format: Attribute_Component_Format This state indicates which components of Attribute 27 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	21:20	Attribute 26 Active Component Format Format: Attribute_Component_Format This state indicates which components of Attribute 26 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.



3DSTATE_SBE			
19:18	<p>Attribute 25 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 25 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
17:16	<p>Attribute 24 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 24 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
15:14	<p>Attribute 23 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 23 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
13:12	<p>Attribute 22 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 22 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
11:10	<p>Attribute 21 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 21 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
9:8	<p>Attribute 20 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 20 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
7:6	<p>Attribute 19 Active Component Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Attribute_Component_Format</td> </tr> </table> <p>This state indicates which components of Attribute 19 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		



3DSTATE_SBE			
5:4	<p>Attribute 18 Active Component Format</p> <table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 18 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
3:2	<p>Attribute 17 Active Component Format</p> <table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 17 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		
1:0	<p>Attribute 16 Active Component Format</p> <table border="1"><tr><td>Format:</td><td>Attribute_Component_Format</td></tr></table> <p>This state indicates which components of Attribute 16 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	Attribute_Component_Format
Format:	Attribute_Component_Format		



3DSTATE_SBE_SWIZ

3DSTATE_SBE_SWIZ			
Source:		RenderCS	
Length 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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		51h 3DSTATE_SBE_SWIZ	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	9h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1..8	15:0	Attribute	
		Format: SF_OUTPUT_ATTRIBUTE_DETAIL[16]	
9..10	63:60	Attribute 15 Wrap Shortest Enables	
	Format: WRAP_SHORTEST_ENABLE		
	59:56	Attribute 14 Wrap Shortest Enables	
	Format: WRAP_SHORTEST_ENABLE		
	55:52	Attribute 13 Wrap Shortest Enables	
		Format: WRAP_SHORTEST_ENABLE	
	51:48	Attribute 12 Wrap Shortest Enables	
		Format: WRAP_SHORTEST_ENABLE	



3DSTATE_SBE_SWIZ		
47:44	Attribute 11 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
43:40	Attribute 10 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
39:36	Attribute 09 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
35:32	Attribute 08 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
31:28	Attribute 07 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
27:24	Attribute 06 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
23:20	Attribute 05 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
19:16	Attribute 04 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
15:12	Attribute 03 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
11:8	Attribute 02 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
7:4	Attribute 01 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE
3:0	Attribute 00 Wrap Shortest Enables	Format: WRAP_SHORTEST_ENABLE



3DSTATE_SCISSOR_STATE_POINTERS

3DSTATE_SCISSOR_STATE_POINTERS			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_SCISSOR_STATE_POINTERS command is used to define the location of the indirect SCISSOR_RECT state.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	0Fh 3DSTATE_SCISSOR_STATE_POINTERS	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h DWORD_COUNT_n	
	Format:	=n	
1	31:5	Scissor Rect Pointer	
		Format: DynamicStateOffset[31:5]SCISSOR_RECT*16 Specifies the 32-byte aligned address offset of the SCISSOR_RECT state. This offset is relative to the Dynamic State Base Address .	
	4:0	Reserved	
		Format: MBZ	



3DSTATE_SF

3DSTATE_SF			
Source:		RenderCS	
Length 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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		13h 3DSTATE_SF	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	2h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31:30	Reserved	
		Format:	MBZ
	29:12	Line Width	
		Format:	U11.7
		Range: [0.0, 2047.9921875]	
		Controls width of line primitives. Setting a Line Width of 0.0 specifies the rasterization 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 -			



3DSTATE_SF			
	<p>zero-width lines are not available when multisampling rasterization is enabled.</p> <p style="text-align: center;">Restriction</p> <p>Line widths higher than 40.0 will appear to be thinner than the programmed line width at certain angles. Software must expose the line widths higher than 40 to APIs only if the loss in quality is acceptable.</p> <p>Software must set the "WM_CHIKEN1" register bits "1 and 3" when rendering the AA lines higher than Line width of "40.0".</p>		
11	<p>Legacy Global Depth Bias Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>Enables the SF to use the Global Depth Offset Constant state unmodified. If this bit is not set, the SF will scale the Global Depth Offset Constant as described in section Error! Reference source not found. of this document.</p> <p style="text-align: center;">Programming Notes</p> <p>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.</p>	Format:	Enable
Format:	Enable		
10	<p>Statistics Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, this FF unit will increment CL_PRIMITIVES_COUNT on behalf of the CLIP stage. If DISABLED, CL_PRIMITIVES_COUNT will be left unchanged.</p> <p style="text-align: center;">Programming Notes</p> <p>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.</p>	Format:	Enable
Format:	Enable		
9:2	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
1	<p>Viewport Transform Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit controls the Viewport Transform function.</p>	Format:	Enable
Format:	Enable		
0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
2	<p>31:29 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ	
	28 Reserved		
	<p>27:18 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
17:16 Line End Cap Antialiasing Region Width			



3DSTATE_SF																			
		<table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field specifies the distances over which the coverage of anti-aliased line end caps are computed.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0.5 pixels</td> <td>0.5 pixels</td> </tr> <tr> <td>1h</td> <td>1.0 pixels</td> <td>1.0 pixels</td> </tr> <tr> <td>2h</td> <td>2.0 pixels</td> <td>2.0 pixels</td> </tr> <tr> <td>3h</td> <td>4.0 pixels</td> <td>4.0 pixels</td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	0.5 pixels	0.5 pixels	1h	1.0 pixels	1.0 pixels	2h	2.0 pixels	2.0 pixels	3h	4.0 pixels	4.0 pixels
	Format:	U2																	
	Value	Name	Description																
	0h	0.5 pixels	0.5 pixels																
	1h	1.0 pixels	1.0 pixels																
	2h	2.0 pixels	2.0 pixels																
	3h	4.0 pixels	4.0 pixels																
	15	Reserved	<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
	Format:	MBZ																	
	14	Reserved	<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																		
13	Reserved																		
12	Reserved																		
11:0	Reserved	<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																		
3	31	<p>Last Pixel Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, the last pixel of a diamond line will be lit. This state will only affect the rasterization of Diamond lines (will not affect wide lines or anti-aliased lines).</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Last pixel is applied to all lines of a LINELIST, and only the last line of a LINESTRIP.</td> </tr> </table>	Format:	Enable	Programming Notes		Last pixel is applied to all lines of a LINELIST, and only the last line of a LINESTRIP.												
	Format:	Enable																	
	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 Provoking Vertex Select	<table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Selects which vertex of a triangle (in a triangle strip or list primitive) is considered the "provoking vertex". Used for flat shading of primitives. Does current implementation send provoking vertex first?</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0</td> </tr> <tr> <td>1h</td> <td>1</td> </tr> <tr> <td>2h</td> <td>2</td> </tr> <tr> <td>3h</td> <td>Reserved</td> </tr> </tbody> </table>	Format:	U2	Value	Name	0h	0	1h	1	2h	2	3h	Reserved					
Format:	U2																		
Value	Name																		
0h	0																		
1h	1																		
2h	2																		
3h	Reserved																		
28:27	Line Strip/List Provoking Vertex Select	<table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Selects which vertex of a line (in a line strip or list primitive) is considered the "provoking vertex".</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0</td> <td>Vertex 0</td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	0	Vertex 0									
Format:	U2																		
Value	Name	Description																	
0h	0	Vertex 0																	



3DSTATE_SF			
	1h	1	Vertex 1
	2h	Reserved	Reserved
	3h	Reserved	Reserved
26:25	Triangle Fan Provoking Vertex Select		
	Format:		U2
	Selects which vertex of a triangle (in a triangle fan primitive) is considered the "provoking vertex".		
	Value	Name	
	0h	0	
	1h	1	
	2h	2	
	3h	Reserved	
24:15	Reserved		
	Format:		MBZ
14	AA Line Distance Mode		
	Format:		U1
	This bit controls the distance computation for antialiased lines.		
	Value	Name	Description
	1h	AALINEDISTANCE_TRUE	True distance computation. This is the normal setting which should yield WHQL compliance.
13	Smooth Point Enable		
	Format:		Enable
	Double Buffer Armed By:		Enables logic to draw smooth OGL Points
	Programming Notes		
	If Enabled, SF will treat points in the same fashion that AA lines are processed		
12	Vertex Sub Pixel Precision Select		
	Format:		U1
	Selects the number of fractional bits maintained in the vertex data		
	Value	Name	Description
	0h	8	8 sub pixel precision bits maintained
	1h	4	4 sub pixel precision bits maintained
	Programming Notes		
	When Conservative Rasterization is enabled, this bit must be programmed to 0.		
11	Point Width Source		
	Controls whether the point width passed on the vertex or from state is used for rendering point		



3DSTATE_SF											
		primitives.									
	<table border="1"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>0h</td><td>Vertex</td><td>Use Point Width on Vertex</td></tr><tr><td>1h</td><td>State [Default]</td><td>Use Point Width from State</td></tr></tbody></table>	Value	Name	Description	0h	Vertex	Use Point Width on Vertex	1h	State [Default]	Use Point Width from State	
Value	Name	Description									
0h	Vertex	Use Point Width on Vertex									
1h	State [Default]	Use Point Width from State									
10:0	Point Width										
	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										



3DSTATE_SO_BUFFER

3DSTATE_SO_BUFFER		
Source:	RenderCS	
Length Bias:	2	
Programming Notes		
Foreach SO Buffer, the 3DSTATE_SO_BUFFER must only be sent once prior to each 3DPRIMITIVE command.		
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: OpCode
	23:16	3D Command Sub Opcode
Default Value: 18h 3DSTATE_SO_BUFFER Format: OpCode		
15:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Default Value: 6h Excludes DWord (0,1) Format: =n	
	Total Length - 2	
1	31	SO Buffer Enable
		Format: Enable
<p>If set, stream output to SO Buffer is enabled, , if 3DSTATE_STREAMOUT::SO Function ENABLE is also enabled..If clear, the SO Buffer is considered "not bound" and effectively treated as a zero-length buffer for the purposes of SO output and overflow detection. If an enabled stream's Stream to Buffer Selects includes this buffer it is by definition an overflow condition. That stream will cause no writes to occur, and only SO_PRIM_STORAGE_NEEDED[<stream>] will increment.</p>		



3DSTATE_SO_BUFFER			
	30:29 SO Buffer Index Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">U2</td></tr></table> Specifies which of the four SO Buffers is being defined.		U2
		U2	
	28:22 SO Buffer Object Control State Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="text-align: center;">MEMORY_OBJECT_CONTROL_STATE</td></tr></table> Specifies the memory object control state for the SO buffer.		MEMORY_OBJECT_CONTROL_STATE
		MEMORY_OBJECT_CONTROL_STATE	
	21 Stream Offset Write Enable Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">Enable</td></tr></table> When set, this field allows the hardware to write SO_WRITE_OFFSET[Buffer#] as specified in the Stream Offset field. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> The field is operates irrespective of whether SO Buffer Enable is set or clear.		Enable
	Enable		
20 Stream Output Buffer Offset Address Enable Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">Enable</td></tr></table> When set, this field allows the hardware to read/write the stream output buffer offset as specified in the "Stream Output Buffer Offset Address" field. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> The field is operates irrespective of whether SO Buffer Enable is set or clear.		Enable	
	Enable		
19:12 Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">MBZ</td></tr></table>		MBZ	
	MBZ		
	11:0 Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">MBZ</td></tr></table>		MBZ
	MBZ		
2..3	63:2 Surface Base Address Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">GraphicsAddress63-2</td></tr></table> This field specifies the starting address of the buffer in Graphics Memory.		GraphicsAddress63-2
		GraphicsAddress63-2	
1:0 Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">MBZ</td></tr></table>		MBZ	
	MBZ		
4	31:30 Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">MBZ</td></tr></table>		MBZ
		MBZ	
29:0 Surface Size Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">U30-1</td></tr></table> This field specifies the size of buffer in number DWords minus 1 of the buffer in Graphics Memory.		U30-1	
	U30-1		



3DSTATE_SO_BUFFER				
5..6	63:2	<p>Stream Output Buffer Offset Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-2</td> </tr> </table> <p>This field specifies the starting address of the buffer in Graphics Memory where the Stream Output Buffer Offset is stored when all the data has been written. It is also used to fetch the stream Output buffer Offset when needed.</p>	Format:	GraphicsAddress63-2
	Format:	GraphicsAddress63-2		
1:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
7	31:0	<p>Stream Offset</p> <p>This field specifies the Offset in stream output buffer to start at, or whether to append to the end of an existing buffer. The Offset must be DWORD aligned. If Stream Offset is equal to 0xFFFFFFFF then load the value at the Stream Output Buffer Offset address into SO_WRITE_OFFSET[Buffer#]. Otherwise, SO_WRITE_OFFSET[Buffer#] = Stream Offset.</p>		



3DSTATE_SO_DECL_LIST

3DSTATE_SO_DECL_LIST												
Source:		RenderCS										
Length 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: OpCode										
	23:16	3D Command Sub Opcode										
		Default Value: 17h 3DSTATE_SO_DECL_LIST Format: OpCode										
	15:9	Reserved										
		Format: MBZ										
8:0	DWord Length											
	Format: =n Total Length - 2											
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[1,257]</td> <td>Excludes DWORD (0,1) 0-128 Entries</td> <td>Value = 2 * (# of SO_DECL quads) + 1</td> </tr> </tbody> </table>	Value	Name	Description	[1,257]	Excludes DWORD (0,1) 0-128 Entries	Value = 2 * (# of SO_DECL quads) + 1					
Value	Name	Description										
[1,257]	Excludes DWORD (0,1) 0-128 Entries	Value = 2 * (# of SO_DECL quads) + 1										
1	31:16	Reserved										
		Format: MBZ										
	15:12	Stream to Buffer Selects [3]										
Format:		U4 bitmask Index of SO Stream										
Identifies to which SO Buffers stream 3 outputs. See Stream To Buffer Selects [0] field description.												
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1xxxb</td> <td>SO Buffer 3</td> </tr> <tr> <td>x1xxb</td> <td>SO Buffer 2</td> </tr> <tr> <td>xx1xb</td> <td>SO Buffer 1</td> </tr> <tr> <td>xxx1b</td> <td>SO Buffer 0</td> </tr> </tbody> </table>	Value	Name	1xxxb	SO Buffer 3	x1xxb	SO Buffer 2	xx1xb	SO Buffer 1	xxx1b	SO Buffer 0
Value	Name											
1xxxb	SO Buffer 3											
x1xxb	SO Buffer 2											
xx1xb	SO Buffer 1											
xxx1b	SO Buffer 0											



3DSTATE_SO_DECL_LIST														
11:8	<p>Stream to Buffer Selects [2]</p> <table border="1"> <tr> <td>Format:</td> <td>U4 bitmask</td> </tr> </table> <p>Identifies to which SO Buffers stream 2 outputs. See Stream To Buffer Selects [0] field description.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>1xxb</td> <td>SO Buffer 3</td> </tr> <tr> <td>x1xb</td> <td>SO Buffer 2</td> </tr> <tr> <td>xx1xb</td> <td>SO Buffer 1</td> </tr> <tr> <td>xxx1b</td> <td>SO Buffer 0</td> </tr> </tbody> </table>	Format:	U4 bitmask	Value	Name	1xxb	SO Buffer 3	x1xb	SO Buffer 2	xx1xb	SO Buffer 1	xxx1b	SO Buffer 0	
	Format:	U4 bitmask												
	Value	Name												
	1xxb	SO Buffer 3												
	x1xb	SO Buffer 2												
	xx1xb	SO Buffer 1												
	xxx1b	SO Buffer 0												
	7:4	<p>Stream to Buffer Selects [1]</p> <table border="1"> <tr> <td>Format:</td> <td>U4 bitmask</td> </tr> </table> <p>Identifies to which SO Buffers stream 1 outputs. See Stream To Buffer Selects [0] field description.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>1xxb</td> <td>SO Buffer 3</td> </tr> <tr> <td>x1xb</td> <td>SO Buffer 2</td> </tr> <tr> <td>xx1xb</td> <td>SO Buffer 1</td> </tr> <tr> <td>xxx1b</td> <td>SO Buffer 0</td> </tr> </tbody> </table>	Format:	U4 bitmask	Value	Name	1xxb	SO Buffer 3	x1xb	SO Buffer 2	xx1xb	SO Buffer 1	xxx1b	SO Buffer 0
		Format:	U4 bitmask											
		Value	Name											
		1xxb	SO Buffer 3											
		x1xb	SO Buffer 2											
xx1xb		SO Buffer 1												
xxx1b		SO Buffer 0												
3:0		<p>Stream to Buffer Selects [0]</p> <table border="1"> <tr> <td>Format:</td> <td>U4 bitmask</td> </tr> </table> <p>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.</p> <p>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_DECLs).</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>1xxb</td> <td>SO Buffer 3</td> </tr> <tr> <td>x1xb</td> <td>SO Buffer 2</td> </tr> <tr> <td>xx1xb</td> <td>SO Buffer 1</td> </tr> <tr> <td>xxx1b</td> <td>SO Buffer 0</td> </tr> </tbody> </table>	Format:	U4 bitmask	Value	Name	1xxb	SO Buffer 3	x1xb	SO Buffer 2	xx1xb	SO Buffer 1	xxx1b	SO Buffer 0
		Format:	U4 bitmask											
		Value	Name											
		1xxb	SO Buffer 3											
		x1xb	SO Buffer 2											
	xx1xb	SO Buffer 1												
	xxx1b	SO Buffer 0												
	2	31:24	<p>Num Entries [3]</p> <table border="1"> <tr> <td>Format:</td> <td>U8 #entries</td> </tr> </table> <p>Specifies the number of valid SO_DECL entries for Stream 3. (See notes in Num Entries [0] field description).</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,128]</td> <td>entries</td> </tr> </tbody> </table>	Format:	U8 #entries	Value	Name	[0,128]	entries					
			Format:	U8 #entries										
			Value	Name										
	[0,128]	entries												



3DSTATE_SO_DECL_LIST										
	23:16	<p>Num Entries [2]</p> <table border="1"> <tr> <td>Format:</td> <td>U8 #entries</td> </tr> <tr> <td colspan="2">Specifies the number of valid SO_DECL entries for Stream 2. (See notes in Num Entries [0] field description).</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>[0,128]</td> <td>entries</td> </tr> </table>	Format:	U8 #entries	Specifies the number of valid SO_DECL entries for Stream 2. (See notes in Num Entries [0] field description).		Value	Name	[0,128]	entries
	Format:	U8 #entries								
	Specifies the number of valid SO_DECL entries for Stream 2. (See notes in Num Entries [0] field description).									
Value	Name									
[0,128]	entries									
15:8	<p>Num Entries [1]</p> <table border="1"> <tr> <td>Format:</td> <td>U8 #entries</td> </tr> <tr> <td colspan="2">Specifies the number of valid SO_DECL entries for Stream 1. (See notes in Num Entries [0] field description).</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>[0,128]</td> <td>entries</td> </tr> </table>	Format:	U8 #entries	Specifies the number of valid SO_DECL entries for Stream 1. (See notes in Num Entries [0] field description).		Value	Name	[0,128]	entries	
Format:	U8 #entries									
Specifies the number of valid SO_DECL entries for Stream 1. (See notes in Num Entries [0] field description).										
Value	Name									
[0,128]	entries									
7:0	<p>Num Entries [0]</p> <table border="1"> <tr> <td>Format:</td> <td>U8 #entries</td> </tr> <tr> <td colspan="2">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(s) 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 streams produce output).</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>[0,128]</td> <td>entries</td> </tr> </table>	Format:	U8 #entries	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(s) 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 streams produce output).		Value	Name	[0,128]	entries	
Format:	U8 #entries									
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(s) 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 streams produce output).										
Value	Name									
[0,128]	entries									
3..n	63:0	<p>Entry</p> <table border="1"> <tr> <td>Format:</td> <td>SO_DECL_ENTRY</td> </tr> </table>	Format:	SO_DECL_ENTRY						
Format:	SO_DECL_ENTRY									



3DSTATE_STENCIL_BUFFER

3DSTATE_STENCIL_BUFFER			
Source:	RenderCS		
Length Bias:	2		
Description			
<p>This command sets the surface state of the separate stencil buffer, delivered as a pipelined state command. However, the state change pipelining isn't completely transparent (see restriction below).</p> <p>WM HW will internally manage the draining pipe and flushing of the caches when this command is issued. The PIPE_CONTROL restrictions are removed.</p>			
Programming Notes			
<p>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).</p> <p>The stencil buffer is always Tile-W</p>			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
		Default Value:	06h 3DSTATE_STENCIL_BUFFER
		Format:	OpCode
	15:8	Reserved	
		Format:	MBZ
7:0	Dword Length	Format:	=n Total Length - 2
	Value	Name	
	3h	Excludes Dword (0,1) [Default]	



3DSTATE_STENCIL_BUFFER								
1	31	Stencil Buffer Enable Format: U1 When set indicates that there is a valid stencil buffer. <div style="text-align: center;">Programming Notes</div> This bit should be "0" if Depth buffer surface format is D16_UNORM OR Depth buffer surface type is NULL.						
	30:29	Reserved Format: MBZ						
	28:22	Stencil Buffer Object Control State Format: MEMORY_OBJECT_CONTROL_STATE Specifies the memory object control state for the stencil buffer.						
	21	Reserved Format: MBZ						
	20:17	Reserved Format: MBZ						
	16:0	Surface Pitch Format: U17-1 Pitch in Bytes This field specifies the pitch of the stencil buffer in (#Bytes - 1). <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>[127, 1FFFFh]</td> <td></td> <td>corresponding to [128B, 128KB]also restricted to a multiple of 128B</td> </tr> </tbody> </table> <div style="text-align: center;">Programming Notes</div> Since this surface is tiled, the pitch specified must be a multiple 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 Overview (vol1a), Memory Data Formats, Surface Layout, 2D Surfaces, Stencil Buffer Layout (section 8.20.4.8).	Value	Name	Description	[127, 1FFFFh]		corresponding to [128B, 128KB]also restricted to a multiple of 128B
	Value	Name	Description					
[127, 1FFFFh]		corresponding to [128B, 128KB]also restricted to a multiple of 128B						
2..3	63:0	Surface Base Address Format: GraphicsAddress[63:0]Stencil_Buffer This field specifies the address of the buffer in Graphics Memory. <div style="text-align: center;">Programming Notes</div> The Stencil Buffer can only be mapped to Main Memory (uncached).						
4	31:15	Reserved Format: MBZ						
	14:0	Surface QPitch Format: QPitch[16:2]						



3DSTATE_STENCIL_BUFFER

Description

This field specifies the distance in rows between array slices. It is used only in the following cases:

- **Surface Array** is enabled *OR*
- **Number of Multisamples** is not NUMSAMPLES_1 and **Multisampled Surface Storage Format** set to MSFMT_MSS *OR*
- **Surface Type** is SURFTYPE_CUBE

The interpretation of this field is dependent on **Surface Type** as follows:

- SURFTYPE_1D: distance in *pixels* between array slices
- SURFTYPE_2D/CUBE: distance in *rows* between array slices
- SURFTYPE_3D: distance in *rows* between R-slices

Value	Name	Description
[4h,1FFFCh]		in multiples of 4 (low 2 bits missing)

Programming Notes

This field must be set to an integer multiple of 8 (QPitch[2] MBZ)

Software must ensure that this field is set to a value sufficiently large such that the array slices in the surface do not overlap. Refer to the Memory Data Formats section for information on how surfaces are stored in memory.



3DSTATE_STREAMOUT

3DSTATE_STREAMOUT			
Source:	RenderCS		
Length Bias:	2		
This command contains pipelined state required by the SOL unit.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	1Eh 3DSTATE_STREAMOUT	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	3h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31	SO Function Enable	
		Format:	U1
<p>If 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.</p> <p>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.</p>			
1	30	API Rendering Disable	
		Format:	U1
<p>If set, Indicates the API wants the SO stage not to forward any topologies down the pipeline.</p> <p>If clear, Indicates the API wants the SO stage to forward topologies associated with Render Stream Select down the pipeline.</p> <p>This bit is used even if SO Function Enable is DISABLED.</p>			



3DSTATE_STREAMOUT		
	Programming Notes	
	The SOL unit generates an SOL_INT::Render_Enable which ultimately controls whether rendering occurs or not.	
29	Reserved	
	Format:	MBZ
28:27	Render Stream Select	
	Format:	U2
	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.	
	SO Function Enable must also be ENABLED in order for this field to select a stream for rendering. When SO Function Enable is DISABLED and Rendering Disable is cleared (i.e., rendering is enabled), StreamID is ignored downstream of the SO stage, allowing any stream to be rendered.	
26	Reorder Mode	
	This bit controls how vertices of triangle objects in TRISTRIP[_ADJ] and TRISTRIP_REV are reordered for the purposes of stream-out only (does not impact rendering). See table in Input Buffering.	
	Value	Name
	Description	
	0h	LEADING
	Reorder the vertices of alternating triangles of a TRISTRIP[_ADJ] such that the leading (first) vertices are in consecutive order starting at v0. A similar reordering is performed on alternating triangles in a TRISTRIP_REV.	
	1h	TRAILING
	Reorder 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:	Enable
	This bit controls whether StreamOutput statistics register(s) can be incremented.	
	Value	Name
	Description	
	0h	Disable
	SO_NUM_PRIMS_WRITTEN[0..3] and SO_PRIM_STORAGE_NEEDED[0..3] registers cannot increment.	
	1h	Enable
	SO_NUM_PRIMS_WRITTEN[0..3] and SO_PRIM_STORAGE_NEEDED[0..3] registers can increment.	



3DSTATE_STREAMOUT																	
	24:23	<p>Force Rendering This field provides a work around override for the computation of SOL_INT::Render_Enable</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Normal</td> <td>SOL_INT::Render_Enable is computed normally</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>2h</td> <td>Force_Off</td> <td>Forces the rendering to be disabled.</td> </tr> <tr> <td>3h</td> <td>Force_on</td> <td>Forces the rendering to be enabled.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Normal	SOL_INT::Render_Enable is computed normally	1h	Reserved		2h	Force_Off	Forces the rendering to be disabled.	3h	Force_on	Forces the rendering to be enabled.
	Value	Name	Description														
	0h	Normal	SOL_INT::Render_Enable is computed normally														
	1h	Reserved															
	2h	Force_Off	Forces the rendering to be disabled.														
	3h	Force_on	Forces the rendering to be enabled.														
22:21	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
20:12	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
11:8	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
7:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
2	31:30	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
	29	<p>Stream 3 Vertex Read Offset</p> <table border="1"> <tr> <td>Format:</td> <td>U1 count of 256-bit units</td> </tr> </table> <p>Specifies amount of data to skip over before reading back Stream 3 vertex data. (See Stream 0 Vertex Read Offset)</p>	Format:	U1 count of 256-bit units													
Format:	U1 count of 256-bit units																
	28:24	<p>Stream 3 Vertex Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U5-1 count of 256-bit units</td> </tr> </table> <p>(See Stream 0 Vertex Read Length)</p>	Format:	U5-1 count of 256-bit units													
Format:	U5-1 count of 256-bit units																
	23:22	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
	21	<p>Stream 2 Vertex Read Offset</p> <table border="1"> <tr> <td>Format:</td> <td>U1 count of 256-bit units</td> </tr> </table> <p>Specifies amount of data to skip over before reading back Stream 2 vertex data. (See Stream 0 Vertex Read Offset)</p>	Format:	U1 count of 256-bit units													
Format:	U1 count of 256-bit units																
	20:16	<p>Stream 2 Vertex Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U5-1 count of 256-bit units</td> </tr> </table>	Format:	U5-1 count of 256-bit units													
Format:	U5-1 count of 256-bit units																
	15:14	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																



3DSTATE_STREAMOUT						
	<p>13 Stream 1 Vertex Read Offset</p> <table border="1"> <tr> <td>Format:</td> <td>U1 count of 256-bit units</td> </tr> </table> <p>Specifies amount of data to skip over before reading back Stream 1 vertex data. (See Stream 0 Vertex Read Offset)</p>	Format:	U1 count of 256-bit units			
	Format:	U1 count of 256-bit units				
	<p>12:8 Stream 1 Vertex Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U5-1 count of 256-bit units</td> </tr> </table> <p>(See Stream 0 Vertex Read Length)</p>	Format:	U5-1 count of 256-bit units			
	Format:	U5-1 count of 256-bit units				
	<p>7:6 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
<p>5 Stream 0 Vertex Read Offset</p> <table border="1"> <tr> <td>Format:</td> <td>U1 count of 256-bit units</td> </tr> </table> <p>Specifies amount of data to skip over before reading back Stream 0 vertex data. Must be zero if the GS is enabled and the Output Vertex Size field in 3DSTATE_GS is programmed to 0 (i.e., one 16B unit).</p>	Format:	U1 count of 256-bit units				
Format:	U1 count of 256-bit units					
<p>4:0 Stream 0 Vertex Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U5-1 count of 256-bit units</td> </tr> </table> <p>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).</p>	Format:	U5-1 count of 256-bit units				
Format:	U5-1 count of 256-bit units					
3	<p>31:28 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ				
	<p>27:16 Buffer 1 Surface Pitch</p>					
	<p>15:12 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
<p>11:0 Buffer 0 Surface Pitch</p> <table border="1"> <tr> <td>Format:</td> <td>U12 pitch in Bytes</td> </tr> </table> <p>This field specifies the pitch of the SO buffer in #Bytes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,2048]</td> <td>Must be 0 or a multiple of 4 Bytes.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p>	Format:	U12 pitch in Bytes	Value	Name	[0,2048]	Must be 0 or a multiple of 4 Bytes.
Format:	U12 pitch in Bytes					
Value	Name					
[0,2048]	Must be 0 or a multiple of 4 Bytes.					



3DSTATE_STREAMOUT		
		A Surface Pitch of 0 indicates an un-bound buffer. No writes are performed. Surface Base Address is ignored.
4	31:28	Reserved Format: MBZ
	27:16	Buffer 3 Surface Pitch Format: U12
	15:12	Reserved Format: MBZ
	11:0	Buffer 2 Surface Pitch Format: U12



3DSTATE_TE

3DSTATE_TE				
Source:	RenderCS			
Length Bias:	2			
The state used by TE is defined with this inline state packet.				
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: 0h 3DSTATE_PIPELINED				
Format: OpCode				
23:16	3D Command Sub Opcode			
	Default Value: 1Ch 3DSTATE_TE			
	Format: OpCode			
15:8	Reserved			
	Format: MBZ			
7:0	DWord Length			
	Format: =n Total Length - 2			
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>2h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>	Value	Name	2h
Value	Name			
2h	Excludes DWord (0,1) [Default]			
1	31:24	Reserved		
		Format: MBZ		
	23:22	Reserved		
		Format: MBZ		
	21	Reserved		
20	Reserved			
19	Reserved			



3DSTATE_TE			
18:17	Reserved		
	Format:	MBZ	
16	Reserved		
	Format:	MBZ	
15:14	Reserved		
	Format:	MBZ	
13:12	Partitioning		
	Format:	U2	
	This field specifies how edges are partitioned based on tessellation factor.		
	Value	Name	Description
	0h	INTEGER	Outside/inside edges are divided into an integer number of equal-sized segments.
	1h	ODD_FRACTIONAL	Outside/inside edges are divided into an odd number of possibly-unequal-sized segments.
2h	EVEN_FRACTIONAL	Outside/inside edges are divided into an even number of possibly-unequal-sized segments.	
11:10	Reserved		
	Format:	MBZ	
9:8	Output Topology		
	Format:	U2	
	This field specifies which primitive types are to be output.		
	Value	Name	Description
	0h	POINT	Points are output (as POINTLIST topologies)
	1h	LINE	Lines are output (as LINESTRIP topologies). Only valid if ISOLINE domain is selected.
2h	TRI_CW	Clockwise-ordered triangles are output (either as TRISTRIP, TRISTRIP_REV or TRILIST topologies). Not valid if ISOLINE domain is selected.	
3h	TRI_CCW	Count-clockwise-ordered triangles are output (either as TRISTRIP, TRISTRIP_REV or TRILIST topologies). Not valid if ISOLINE domain is selected.	
7:6	Reserved		
	Format:	MBZ	
5:4	TE Domain		
	Format:	U2	
	This field specifies which type of domain is to be tessellated.		
	Value	Name	Description
	0h	QUAD	2D (U, V) domain is tessellated
1h	TRI	Triangular (U, V, W) domain is tessellated	



3DSTATE_TE		
	2h	ISOLINE 2D (U, V) domain is tessellated.
3	Reserved	
	Format:	MBZ
2:1	TE Mode	
	Format:	U2
	When TE Enable is ENABLED, this field specifies the overall operation of the TE stage. This field is ignored if TE Enable is DISABLED.	
	Value	Name Description
	0h	HW_TESS Normal HW Tessellation Mode. The TessFactors are read from the patch URB entry, and are used to perform fixed-function hardware tessellation of the specified domain.
0	TE Enable	
	Format:	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	Maximum Tessellation Factor Odd
	Format:	IEEE_Float
	This field specifies the maximum TessFactor for ODD_FRACTIONAL partitioning when in HW_TESS mode.	
	Value	Name Description
	[427c0000h,427c0000h]	63 [Default] Per API Spec, For normal operation software should set this value to 63.0
	Programming Notes	
	Note that ISOLINE's LineDensity TF is always subjected to INTEGER partitioning regardless of the Partitioning state.	
3	31:0	Maximum Tessellation Factor Not Odd
	Format:	IEEE_Float
	This field specifies the maximum TessFactor for EVEN_FRACTIONAL or INTEGER partitioning when in HW_TESS mode.	
	Value	Name Description
	[42800000h,42800000h]	64 [Default] Per API Spec, For normal operation software should set this value to 64.0



3DSTATE_TE

Programming Notes

Note that ISOLINE's LineDensity TF is always subjected to INTEGER partitioning regardless of the Partitioning state.



3DSTATE_URB_CLEAR

3DSTATE_URB_CLEAR			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_URB_CLEAR command allows SW to clear (write zero) to a section in the URB.			
Programming Notes			
<ul style="list-style-type: none"> The command temporarily halts command execution. This command is not a part of context save/restore. 			
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:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		1Dh 3DSTATE_URB_CLEAR	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:30	Reserved	
		Format:	MBZ
	29:16	URB Clear Length This field specifies the number of 256b entries in the URB to be cleared to zero.	
15	Reserved		
	Format:	MBZ	



3DSTATE_URB_CLEAR		
14:0	URB Address Format: <table border="1"><tr><td>URBAddress[19:5] 256b aligned</td></tr></table> This field specifies Bits 19:5 of the URB Address	URBAddress[19:5] 256b aligned
URBAddress[19:5] 256b aligned		



3DSTATE_URB_DS

3DSTATE_URB_DS			
Source:	RenderCS		
Length Bias:	2		
Description			
<p>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.</p> <p>The URB Starting Address and Number of URB Entries fields shall be programmed as if there is only one slice enabled. When more than one slice is enabled, hardware will (a) recompute the actual URB Starting Address based on the number of enabled slices and (b) multiply the Number of URB Entries by the number of enabled slices. Software shall ensure that the values programmed do not exceed the URB capacity of a single slice. Refer to the L3 allocation and programming guide for valid URB configurations.</p>			
Programming Notes			
<p>When programming DS URB state for the RCS 3D pipe, 3DSTATE_URB_VS, 3DSTATE_URB_HS, and 3DSTATE_URB_GS must also be programmed in order for the programming of this state to be valid.</p>			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
		Default Value:	32h 3DSTATE_URB_DS
		Format:	OpCode
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h DWORD_COUNT_n	
	Format:	=n	



3DSTATE_URB_DS													
1	31:25	<p>DS URB Starting Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U7</td> </tr> </table> <p>Offset from the start of the URB memory where DS starts its allocation, specified in multiples of 8 KB.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,48]</td> <td></td> <td>Device[SliceCount] == 1</td> </tr> <tr> <td>[4,48]</td> <td></td> <td>Device[SliceCount] GT 1</td> </tr> </tbody> </table>	Format:	U7	Value	Name	Exists If	[0,48]		Device[SliceCount] == 1	[4,48]		Device[SliceCount] GT 1
	Format:	U7											
	Value	Name	Exists If										
[0,48]		Device[SliceCount] == 1											
[4,48]		Device[SliceCount] GT 1											
24:16	<p>DS URB Entry Allocation Size</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U9-1 Count of 512-bit units</td> </tr> </table> <p>Specifies the length of each URB entry owned by DS. This field is always used (even if DS Function Enable is DISABLED).</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,9]</td> <td></td> </tr> </tbody> </table>	Format:	U9-1 Count of 512-bit units	Value	Name	[0,9]							
Format:	U9-1 Count of 512-bit units												
Value	Name												
[0,9]													
15:0	<p>DS Number of URB Entries</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td> <p>Specifies the number of URB entries that are used by DS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below.</p> <p>This field is always used (even if DS Function Enable is DISABLED).</p> <p>If Domain Shader Thread Dispatch is Enabled then the minimum number of handles that must be allocated is 34 URB entries.</p> </td> </tr> </tbody> </table> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,1120]</td> <td></td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td> <p>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"</p> </td> </tr> </tbody> </table>	Description	<p>Specifies the number of URB entries that are used by DS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below.</p> <p>This field is always used (even if DS Function Enable is DISABLED).</p> <p>If Domain Shader Thread Dispatch is Enabled then the minimum number of handles that must be allocated is 34 URB entries.</p>	Value	Name	[0,1120]		Programming Notes	<p>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"</p>				
Description													
<p>Specifies the number of URB entries that are used by DS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below.</p> <p>This field is always used (even if DS Function Enable is DISABLED).</p> <p>If Domain Shader Thread Dispatch is Enabled then the minimum number of handles that must be allocated is 34 URB entries.</p>													
Value	Name												
[0,1120]													
Programming Notes													
<p>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"</p>													



3DSTATE_URB_GS

3DSTATE_URB_GS			
Source:	RenderCS		
Length Bias:	2		
Description			
<p>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.</p> <p>The URB Starting Address and Number of URB Entries fields shall be programmed as if there is only one slice enabled. When more than one slice is enabled, hardware will (a) recompute the actual URB Starting Address based on the number of enabled slices and (b) multiply the Number of URB Entries by the number of enabled slices. Software shall ensure that the values programmed do not exceed the URB capacity of a single slice. Refer to the L3 allocation and programming guide for valid URB configurations</p>			
Programming Notes			
<p>When programming GS URB state for the RCS 3D pipe, 3DSTATE_URB_VS, 3DSTATE_URB_HS, and 3DSTATE_URB_DS must also be programmed in order for the programming of this state to be valid.</p>			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
		Default Value:	33h 3DSTATE_URB_GS
		Format:	OpCode
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h DWORD_COUNT_n	
	Format:	=n	



3DSTATE_URB_GS														
1	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">31:25</td> <td> GS URB Starting Address Format: U7 Offset from the start of the URB memory where GS starts its allocation, specified in multiples of 8 KB. </td> </tr> <tr> <td></td> <td> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 60%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,48]</td> <td></td> <td>Device[SliceCount] == 1</td> </tr> <tr> <td>[4,48]</td> <td></td> <td>Device[SliceCount] GT 1</td> </tr> </tbody> </table> </td> </tr> </table>	31:25	GS URB Starting Address Format: U7 Offset from the start of the URB memory where GS starts its allocation, specified in multiples of 8 KB.		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 60%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,48]</td> <td></td> <td>Device[SliceCount] == 1</td> </tr> <tr> <td>[4,48]</td> <td></td> <td>Device[SliceCount] GT 1</td> </tr> </tbody> </table>	Value	Name	Exists If	[0,48]		Device[SliceCount] == 1	[4,48]		Device[SliceCount] GT 1
31:25	GS URB Starting Address Format: U7 Offset from the start of the URB memory where GS starts its allocation, specified in multiples of 8 KB.													
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 60%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,48]</td> <td></td> <td>Device[SliceCount] == 1</td> </tr> <tr> <td>[4,48]</td> <td></td> <td>Device[SliceCount] GT 1</td> </tr> </tbody> </table>	Value	Name	Exists If	[0,48]		Device[SliceCount] == 1	[4,48]		Device[SliceCount] GT 1				
Value	Name	Exists If												
[0,48]		Device[SliceCount] == 1												
[4,48]		Device[SliceCount] GT 1												
	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">24:16</td> <td> GS URB Entry Allocation Size Format: U9-1 512-bit units Specifies the length of each URB entry owned by GS. This field is always used (even if GS Function Enable is DISABLED). </td> </tr> </table>	24:16	GS URB Entry Allocation Size Format: U9-1 512-bit units Specifies the length of each URB entry owned by GS. This field is always used (even if GS Function Enable is DISABLED).											
24:16	GS URB Entry Allocation Size Format: U9-1 512-bit units Specifies the length of each URB entry owned by GS. This field is always used (even if GS Function Enable is DISABLED).													
	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">15:0</td> <td> GS Number of URB Entries Specifies the number of URB entries that are used by GS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below. This field is always used (even if GS Function Enable is DISABLED). </td> </tr> <tr> <td></td> <td style="text-align: center;"> Programming Notes </td> </tr> <tr> <td></td> <td> 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. </td> </tr> <tr> <td></td> <td> GS Number of URB Entries must be divisible by 8 if the GS URB Entry Allocation Size is less than 9 512-bit URB entries. "2:0" = reserved "000" </td> </tr> <tr> <td></td> <td> When 3DSTATE_GS:Enable is true, the minimum number of GS Number of URB Entries must be set to 2. </td> </tr> </table>	15:0	GS Number of URB Entries Specifies the number of URB entries that are used by GS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below. This field is always used (even if GS Function Enable is DISABLED).		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 must be divisible by 8 if the GS URB Entry Allocation Size is less than 9 512-bit URB entries. "2:0" = reserved "000"		When 3DSTATE_GS:Enable is true, the minimum number of GS Number of URB Entries must be set to 2.			
15:0	GS Number of URB Entries Specifies the number of URB entries that are used by GS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below. This field is always used (even if GS Function Enable is DISABLED).													
	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 must be divisible by 8 if the GS URB Entry Allocation Size is less than 9 512-bit URB entries. "2:0" = reserved "000"													
	When 3DSTATE_GS:Enable is true, the minimum number of GS Number of URB Entries must be set to 2.													



3DSTATE_URB_HS

3DSTATE_URB_HS		
Source:	RenderCS	
Length Bias:	2	
Description		
<p>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.</p> <p>The URB Starting Address and Number of URB Entries fields shall be programmed as if there is only one slice enabled. When more than one slice is enabled, hardware will (a) recompute the actual URB Starting Address based on the number of enabled slices and (b) multiply the Number of URB Entries by the number of enabled slices. Software shall ensure that the values programmed do not exceed the URB capacity of a single slice. Refer to the L3 allocation and programming guide for valid URB configurations</p>		
Programming Notes		
<p>When programming HS URB state for the RCS 3D pipe, 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.</p>		
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: 0h 3DSTATE_PIPELINED Format: OpCode
	23:16	3D Command Sub Opcode
Default Value: 31h 3DSTATE_URB_HS Format: OpCode		
15:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Default Value: 0h DWORD_COUNT_n Format: =n	
1	31:25	HS URB Starting Address
		Format: U7



3DSTATE_URB_HS											
		<p>Offset from the start of the URB memory where HS starts its allocation, specified in multiples of 8 KB.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,48]</td> <td></td> <td>Device[SliceCount] == 1</td> </tr> <tr> <td>[4,48]</td> <td></td> <td>Device[SliceCount] GT 1</td> </tr> </tbody> </table>	Value	Name	Exists If	[0,48]		Device[SliceCount] == 1	[4,48]		Device[SliceCount] GT 1
Value	Name	Exists If									
[0,48]		Device[SliceCount] == 1									
[4,48]		Device[SliceCount] GT 1									
24:16	<p>HS URB Entry Allocation Size</p> <table border="1"> <tr> <td>Format:</td> <td>U9-1 Count of 512-bit units</td> </tr> </table> <p>Specifies the length of each URB entry owned by HS. This field is always used (even if HS Function Enable is DISABLED).</p>		Format:	U9-1 Count of 512-bit units							
Format:	U9-1 Count of 512-bit units										
15:0	<p>HS Number of URB Entries</p> <p>Specifies the number of URB entries that are used by HS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below.</p> <p>This field is always used (even if HS Function Enable is DISABLED).</p> <p>Programming Restriction:HS Number of URB Entries must be divisible by 8 if the HS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000"</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,672]</td> <td></td> </tr> <tr> <td>[0,128]</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>When 3DSTATE_HS:Enable is true and 3DSTATE_HS:Dispatch Mode is set to 8 patch, the minimum number of HS Number of URB Entries must be set to 16. When 3DSTATE_HS:Enable is true and 3DSTATE_HS:Dispatch Mode is set to dual patch, the minimum number of HS Number of URB Entries must be set to 4.</td> </tr> </tbody> </table>		Value	Name	[0,672]		[0,128]		Programming Notes	When 3DSTATE_HS:Enable is true and 3DSTATE_HS:Dispatch Mode is set to 8 patch, the minimum number of HS Number of URB Entries must be set to 16. When 3DSTATE_HS:Enable is true and 3DSTATE_HS:Dispatch Mode is set to dual patch, the minimum number of HS Number of URB Entries must be set to 4.	
Value	Name										
[0,672]											
[0,128]											
Programming Notes											
When 3DSTATE_HS:Enable is true and 3DSTATE_HS:Dispatch Mode is set to 8 patch, the minimum number of HS Number of URB Entries must be set to 16. When 3DSTATE_HS:Enable is true and 3DSTATE_HS:Dispatch Mode is set to dual patch, the minimum number of HS Number of URB Entries must be set to 4.											



3DSTATE_URB_VS

3DSTATE_URB_VS			
Source:	RenderCS, PositionCS		
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.			
The offset and size should be programmed as if there is only one slice enabled. Hardware will grow the size based on the slice configuration. Software shall ensure that the values programmed do not exceed the URB capacity of one slice. Refer to the L3 allocation and programming guide for valid URB configurations.			
Programming Notes			
When programming VS URB state for the RCS 3D pipe, 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:	OpCode
	28:27	Command SubType	
		Default Value:	3h GFXPIPE_3D
		Format:	OpCode
	26:24	3D Command Opcode	
Default Value:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	30h 3DSTATE_URB_VS	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h DWORD_COUNT_n	
	Format:	=n	
1	31:25	VS URB Starting Address	
		Format:	U7



3DSTATE_URB_VS											
	<p>Offset from the start of the URB memory where VS starts its allocation, specified in multiples of 8 KB.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,48]</td> <td></td> <td>Device[SliceCount] == 1</td> </tr> <tr> <td>[4,48]</td> <td></td> <td>Device[SliceCount] GT 1</td> </tr> </tbody> </table>	Value	Name	Exists If	[0,48]		Device[SliceCount] == 1	[4,48]		Device[SliceCount] GT 1	
Value	Name	Exists If									
[0,48]		Device[SliceCount] == 1									
[4,48]		Device[SliceCount] GT 1									
24:16	<p>VS URB Entry Allocation Size</p> <table border="1"> <tr> <td>Format:</td> <td>U9-1 count of 512-bit units</td> </tr> </table> <p>Specifies the length of each URB entry owned by VS. This field is always used (even if VS Function Enable is DISABLED).</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>Programming Restriction: As the VS URB entry serves as both the per-vertex input and output of the VS shader, the VS URB Allocation Size must be sized to the maximum of the vertex input and output structures.</td> </tr> </tbody> </table>	Format:	U9-1 count of 512-bit units	Programming Notes	Programming Restriction: As the VS URB entry serves as both the per-vertex input and output of the VS shader, the VS URB Allocation Size must be sized to the maximum of the vertex input and output structures.						
Format:	U9-1 count of 512-bit units										
Programming Notes											
Programming Restriction: As the VS URB entry serves as both the per-vertex input and output of the VS shader, the VS URB Allocation Size must be sized to the maximum of the vertex input and output structures.											
15:0	<p>VS Number of URB Entries</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>Specifies the number of URB entries that are used by VS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below. This field is always used (even if VS Function Enable is DISABLED).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[64,1856]</td> <td></td> </tr> <tr> <td>[34,352]</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>Programming Restriction: VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"</td> </tr> </tbody> </table>	Format:	U16	Value	Name	[64,1856]		[34,352]		Programming Notes	Programming Restriction: VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"
Format:	U16										
Value	Name										
[64,1856]											
[34,352]											
Programming Notes											
Programming Restriction: VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"											



3DSTATE_VERTEX_BUFFERS

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).			
The VF cache (and the VFR cache if POSH is enabled) needs to be invalidated before binding and then using Vertex Buffers that overlap with any previously bound Vertex Buffer (at a 64B granularity) since the last invalidation.			
A VF cache invalidate is performed by setting the "VF Cache Invalidation Enable" bit in PIPE_CONTROL.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	03h GFXPIPE
		Format:	Opcode
	28:27	Command SubType	
		Default Value:	3h GFXPIPE_3D
		Format:	Opcode
	26:24	3D Command Opcode	
		Default Value:	0h 3DSTATE_PIPELINED
		Format:	Opcode



3DSTATE_VERTEX_BUFFERS						
	23:16	3D Command Sub Opcode				
		Default Value: 08h 3DSTATE_VERTEX_BUFFERS				
	Format: Opcode					
	15:8	Reserved				
	7:0	DWord Length				
		Format: =n				
n = 4b-1 (where b = # of buffer states included)						
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">3</td> <td>DWORD_COUNT_n [Default]</td> </tr> <tr> <td style="text-align: center;">[3,131]</td> <td>1-33 Buffers</td> </tr> </tbody> </table>		Value	Name	3	DWORD_COUNT_n [Default]	[3,131]
Value	Name					
3	DWORD_COUNT_n [Default]					
[3,131]	1-33 Buffers					
1..n	127:0	Vertex Buffer State				
Format: VERTEX_BUFFER_STATE						



3DSTATE_VERTEX_ELEMENTS

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 3DPRIMITIVE 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 GFXPIPE
		Format:	Opcode
	28:27	Command SubType	
		Default Value:	3h 3D
		Format:	Opcode
	26:24	3D Command Opcode	
		Default Value:	0h 3DSTATE_PIPELINED
		Format:	Opcode



3DSTATE_VERTEX_ELEMENTS												
	23:16	3D Command Sub Opcode <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Default Value:</td> <td>09h 3DSTATE_VERTEX_ELEMENTS</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	09h 3DSTATE_VERTEX_ELEMENTS	Format:	Opcode						
	Default Value:	09h 3DSTATE_VERTEX_ELEMENTS										
	Format:	Opcode										
	15:8	Reserved										
	7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>=n</td> </tr> </table> <p>Vertex Element Count = (DWord Count + 1) / 2</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 55%;">Name</th> <th style="width: 30%;">Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>DWORD_COUNT_n [Default]</td> <td>excludes DWords 0,1</td> </tr> <tr> <td>[1,67]</td> <td>Range</td> <td>1-34 Elements</td> </tr> </tbody> </table>	Format:	=n	Value	Name	Description	1	DWORD_COUNT_n [Default]	excludes DWords 0,1	[1,67]	Range
Format:	=n											
Value	Name	Description										
1	DWORD_COUNT_n [Default]	excludes DWords 0,1										
[1,67]	Range	1-34 Elements										
1..n	63:0	Element <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>VERTEX_ELEMENT_STATE</td> </tr> </table>	Format:	VERTEX_ELEMENT_STATE								
Format:	VERTEX_ELEMENT_STATE											



3DSTATE_VF_COMPONENT_PACKING

3DSTATE_VF_COMPONENT_PACKING			
Source:	RenderCS		
Length Bias:	2		
<p>This command is used to specify, separately for Vertex Elements [0-31], which post-conversion, 32-bit components are "enabled" to be stored in the URB , and which are "disabled" (not stored). 128 per-component enable bits are provided. Disabling all four components for a given Vertex Element will result in no data stored for that element. Note that any insertion of SGVs (3DSTATE_VF_SGVS) is performed before the packing operation.</p> <p>The Component Packing Enable bit (3DSTATE_VF) controls the overall packing process. If that bit is set, the packing process is enabled and the bit mask provided in this command is used to control which components are stored. If that bit is clear, the packing process is disabled - all four components of "valid" Vertex Elements will be stored.</p>			
Programming Notes			
Programming Restrictions:			
<ul style="list-style-type: none"> The Vertex Elements referenced in this command correspond to the first 32 VERTEX_ELEMENT structures passed in 3DSTATE_VERTEX_ELEMENTS. A Vertex Element must be marked as "Valid" via 3DSTATE_VERTEX_ELEMENTS or be an SGV or an element between the last valid element and the last SGV in order for the corresponding Component Enable bits of this command to be utilized. No enable bits are provided for Vertex Elements [32-33], and therefore no packing is performed on these elements (if Valid, all 4 components are stored). If a Vertex Element has Edge Flag Enable set no packing is performed for that element and the corresponding packing state is ignored. Component packing is probably only useful for SIMD8 VS thread execution. 			
At least one component of one "valid" Vertex Element must be enabled.			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode



3DSTATE_VF_COMPONENT_PACKING		
	23:16	3D Command Sub Opcode Default Value: 55h 3DSTATE_VF_COMPONENT_PACKING Format: OpCode
	15:8	Reserved Format: MBZ
	7:0	DWord Length Default Value: 3h Excludes DWord (0,1) Format: =n Total Length - 2
1	31:28	Vertex Element 07 Enables Format: COMPONENT_ENABLES
	27:24	Vertex Element 06 Enables Format: COMPONENT_ENABLES
	23:20	Vertex Element 05 Enables Format: COMPONENT_ENABLES
	19:16	Vertex Element 04 Enables Format: COMPONENT_ENABLES
	15:12	Vertex Element 03 Enables Format: COMPONENT_ENABLES
	11:8	Vertex Element 02 Enables Format: COMPONENT_ENABLES
	7:4	Vertex Element 01 Enables Format: COMPONENT_ENABLES
	3:0	Vertex Element 00 Enables Format: COMPONENT_ENABLES
2	31:28	Vertex Element 15 Enables Format: COMPONENT_ENABLES
	27:24	Vertex Element 14 Enables Format: COMPONENT_ENABLES
	23:20	Vertex Element 13 Enables Format: COMPONENT_ENABLES
	19:16	Vertex Element 12 Enables Format: COMPONENT_ENABLES
	15:12	Vertex Element 11 Enables Format: COMPONENT_ENABLES



3DSTATE_VF_COMPONENT_PACKING		
	11:8	Vertex Element 10 Enables Format: COMPONENT_ENABLES
	7:4	Vertex Element 09 Enables Format: COMPONENT_ENABLES
	3:0	Vertex Element 08 Enables Format: COMPONENT_ENABLES
3	31:28	Vertex Element 23 Enables Format: COMPONENT_ENABLES
	27:24	Vertex Element 22 Enables Format: COMPONENT_ENABLES
	23:20	Vertex Element 21 Enables Format: COMPONENT_ENABLES
	19:16	Vertex Element 20 Enables Format: COMPONENT_ENABLES
	15:12	Vertex Element 19 Enables Format: COMPONENT_ENABLES
	11:8	Vertex Element 18 Enables Format: COMPONENT_ENABLES
	7:4	Vertex Element 17 Enables Format: COMPONENT_ENABLES
	3:0	Vertex Element 16 Enables Format: COMPONENT_ENABLES
4	31:28	Vertex Element 31 Enables Format: COMPONENT_ENABLES
	27:24	Vertex Element 30 Enables Format: COMPONENT_ENABLES
	23:20	Vertex Element 29 Enables Format: COMPONENT_ENABLES
	19:16	Vertex Element 28 Enables Format: COMPONENT_ENABLES
	15:12	Vertex Element 27 Enables Format: COMPONENT_ENABLES
	11:8	Vertex Element 26 Enables Format: COMPONENT_ENABLES



3DSTATE_VF_COMPONENT_PACKING		
	7:4	Vertex Element 25 Enables
		Format: COMPONENT_ENABLES
	3:0	Vertex Element 24 Enables
		Format: COMPONENT_ENABLES



3DSTATE_VF

3DSTATE_VF			
Source:	RenderCS		
Length Bias:	2		
Description			
This command is used to set various state variables in the VF stage.			
The use of the component packing mask is specified via 3DSTATE_VF_COMPONENT_PACKING			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
		Default Value:	0Ch 3DSTATE_VF
		Format:	OpCode
15	Reserved	Format: MBZ	
14	Reserved	Format: MBZ	
13	Reserved	Format: MBZ	
12	Reserved		
11	Reserved	Format: MBZ	
10	Sequential Draw Cut Index Enable	Format: Enable	
	If ENABLED, vertex indices in SEQUENTIAL 3DPRIMITIVE commands are compared to the Cut Index (specified below). When the vertex index matches the Cut Index any previous topology is terminated. If DISABLED, vertex indices are not compared to the Cut Index. This field can only be enabled for		



3DSTATE_VF					
	certain primitive topology types. Refer to the table later in this section for details.				
9	<p>Component Packing Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, vertex element component packing (as specified by 3DSTATE_VF_COMPONENT_PACKING) is performed before vertices are written into the URB. If DISABLED, no component packing is performed - all components of valid vertex elements will be stored in the URB.</p>	Format:	Enable		
Format:	Enable				
8	<p>Indexed Draw Cut Index Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, vertex indices in RANDOM 3DPRIMITIVE commands are compared to the Cut Index (specified below). When the vertex index matches the Cut Index any previous topology is terminated. If DISABLED, vertex indices are not compared to the Cut Index and are used strictly as indices into vertex buffers. This field can only be enabled for certain primitive topology types. Refer to the table later in this section for details.</p>	Format:	Enable		
Format:	Enable				
7:0	<p>DWord Length</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table>	Default Value:	0h Excludes DWord (0,1)	Format:	=n Total Length - 2
Default Value:	0h Excludes DWord (0,1)				
Format:	=n Total Length - 2				
1	<p>31:0 Cut Index</p> <p>This field specifies the index value that is considered the "cut index" which vertex indices are compared to if a Cut Index Enable is set. The Cut Index is compared to the fetched (and possibly-sign-extended) vertex index, and if these values are equal, the current primitive topology is terminated. Note that, for index buffers less than 32bpp, it is possible to set the Cut Index to a (large) value that will never match a sign-extended vertex index.</p>				



3DSTATE_VF_INSTANCING

3DSTATE_VF_INSTANCING								
Source:	RenderCS							
Length Bias:	2							
This command is used to control the "instancing" state associated with a specific vertex element.								
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:	0h 3DSTATE_PIPELINED					
		Format:	OpCode					
	23:16	3D Command Sub Opcode						
Default Value:		49h 3DSTATE_VF_INSTANCING						
Format:		OpCode						
15:8	Reserved							
	Format:	MBZ						
7:0	DWord Length	Format:	=n Total Length - 2					
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> <tr> <td>43h</td> <td>Context Restore</td> </tr> </tbody> </table>	Value	Name	1h	Excludes DWord (0,1) [Default]	43h	Context Restore
	Value	Name						
	1h	Excludes DWord (0,1) [Default]						
	43h	Context Restore						
1	31:9	Reserved						
		Format:	MBZ					
		8	Instancing Enable					
		Format:	Enable					
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disabled</td> <td>This vertex element is not instanced and therefore vertices within instances can each receive different data for this vertex element. Within each instance, the source vertex data for this vertex element is determined according the the Vertex Access Type of the 3DPRIMITIVE command. Instance Data Step Rate is</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disabled	This vertex element is not instanced and therefore vertices within instances can each receive different data for this vertex element. Within each instance, the source vertex data for this vertex element is determined according the the Vertex Access Type of the 3DPRIMITIVE command. Instance Data Step Rate is
Value	Name	Description						
0h	Disabled	This vertex element is not instanced and therefore vertices within instances can each receive different data for this vertex element. Within each instance, the source vertex data for this vertex element is determined according the the Vertex Access Type of the 3DPRIMITIVE command. Instance Data Step Rate is						



3DSTATE_VF_INSTANCING			
			ignored for this vertex element.
	1h	Enabled	This vertex element is instanced and therefore vertices within instances will receive the same data for this vertex element. The source pointer for this particular vertex element will be (a) initialized at the start of 3DPRIMITIVE processing, (b) held constant for all vertices within an instance, and (c) advanced between instances as a function of Instance Data Step Rate.
	7:6	Reserved	
		Format:	MBZ
	5:0	Vertex Element Index	
		Format:	U6
		This field identifies which vertex element state is to be modified by this command.	
		Value	Name
		[0,33]	
2	31:0	Instance Data Step Rate If Instancing Enable is ENABLED, this field determines the rate at which data for this particular vertex element is changed between instances. Only after the number of instances specified by this field is generated is new (sequential) vertex element data provided. This process continues for each group of instances defined in the 3DPRIMITIVE command. For example, a value of 1 in this field causes new data to be supplied for this vertex element with each sequential (instance) group of vertices. A value of 2 causes every other instance group of vertices to be provided with new vertex element data. The special value of 0 causes all vertices of all instances generated by the 3DPRIMITIVE command to be provided with the same data for this vertex element. (The same effect can be achieved by setting this field to its maximum value.) If Instancing Enable is DISABLED, this field is ignored.	



3DSTATE_VF_SGVS

3DSTATE_VF_SGVS			
Source:	RenderCS		
Length Bias:	2		
Description			
<p>This command is used to control the insertion of the VertexID and InstanceID System-Generated Values (SGVs) into an input Vertex URB Entry (VUE) (available as input to a VS thread). VertexID and InstanceID insertion can be individually controlled. The insertion locations are specified as 128-bit element locations (starting at the beginning of the VUE) and the 32-bit component within those specified elements. The SGV values can be inserted either (a) within a valid vertex element (in which case the value overwrites the value specified via 3DSTATE_VERTEX_ELEMENTS) or (b) beyond the last valid vertex element written to the URB. This permits some orthogonality between the programming of vertex elements (which typically is known at draw time) and programming of SGV insertion (which is associated with the shader). There are some restrictions however (see below). If an SGV is inserted beyond the last valid vertex element, zeroes are first inserted in the VUE after the last valid vertex element up to and including the vertex element receiving an SGV. If both of the SGVs are enabled for insertion, the zeroes will extend to the last (largest index) vertex element receiving an SGV. Then the SGV(s) are inserted.</p> <p>The insertion of SGV values occurs before any component packing (3DSTATE_VF_COMPONENT_PACKING). Therefore the Element Offsets and Component Numbers specified in this command refer to the pre-packed data, following 3DSTATE_VERTEX_ELEMENT processing.</p>			
Programming Notes			
<p>Programming Restrictions:</p> <ul style="list-style-type: none"> It is INVALID to store both the VertexID and InstanceID in the same element/component location within the VUE. The states programmed by this command overwrite the state programmed by any previous commands. I.e., VertexID and InstanceID (if enabled) can only be inserted in one component of a vertex. It is INVALID to insert an SGV value past the end of the VUE entry (as determined by VS URB Entry Allocation Size) or past the 33rd vertex element. Therefore the programming of VS URB Entry Allocation Size needs to comprehend any SGV insertion requirements. It is INVALID to use this command to overwrite any portion of a 64-bit vertex element component. It is INVALID to use this command to overwrite a EdgeFlag vertex element component or any vertex element beyond it. 			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	OpCode
	28:27	Command SubType	
		Default Value:	3h GFXPIPE_3D



3DSTATE_VF_SGVS																
		Format: _____ OpCode														
	26:24	3D Command Opcode														
		Default Value: _____ 0h 3DSTATE_PIPELINED														
		Format: _____ OpCode														
	23:16	3D Command Sub Opcode														
		Default Value: _____ 4Ah 3DSTATE_VF_SGVS														
15:8	Reserved															
	Format: _____ MBZ															
7:0	DWord Length															
	Default Value: _____ 0h Excludes DWord (0,1)															
	Format: _____ =n Total Length - 2															
1	31	InstanceID Enable														
		Format: _____ Enable														
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disabled</td> <td>InstanceID is not inserted</td> </tr> <tr> <td>1h</td> <td>Enabled</td> <td>InstanceID is inserted</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disabled	InstanceID is not inserted	1h	Enabled	InstanceID is inserted					
		Value	Name	Description												
	0h	Disabled	InstanceID is not inserted													
	1h	Enabled	InstanceID is inserted													
	30:29	InstanceID Component Number														
		If InstanceID Enable is ENABLED, this field specifies the 32-bit component location (within the 4-component VUE) where it is inserted. If InstanceID Enable is DISABLED, this field is ignored.														
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 75%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>COMP_0</td> <td>If enabled, InstanceID is inserted in component 0 (.x)</td> </tr> <tr> <td>1</td> <td>COMP_1</td> <td>If enabled, InstanceID is inserted in component 1 (.y)</td> </tr> <tr> <td>2</td> <td>COMP_2</td> <td>If enabled, InstanceID is inserted in component 2 (.z)</td> </tr> <tr> <td>3</td> <td>COMP_3</td> <td>If enabled, InstanceID is inserted in component 3 (.w)</td> </tr> </tbody> </table>	Value	Name	Description	0	COMP_0	If enabled, InstanceID is inserted in component 0 (.x)	1	COMP_1	If enabled, InstanceID is inserted in component 1 (.y)	2	COMP_2	If enabled, InstanceID is inserted in component 2 (.z)	3	COMP_3
Value		Name	Description													
0		COMP_0	If enabled, InstanceID is inserted in component 0 (.x)													
1	COMP_1	If enabled, InstanceID is inserted in component 1 (.y)														
2	COMP_2	If enabled, InstanceID is inserted in component 2 (.z)														
3	COMP_3	If enabled, InstanceID is inserted in component 3 (.w)														
28:22	Reserved															
	Format: _____ MBZ															
21:16	InstanceID Element Offset															
	Format: _____ U6 Offset of 128-bit element															
	If InstanceID Enable is ENABLED, this field specifies the VUE element offset of the 128-bit element where it is to be inserted. The InstanceID Component Number specifies where in the specified element it is inserted.															
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,33]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,33]												
Value	Name															
[0,33]																



3DSTATE_VF_SGVS		
15	VertexID Enable	
	Format: Enable	
	Value	Name Description
	0h	Disabled VertexID is not inserted
	1h	Enabled VertexID is inserted
14:13	VertexID Component Number	
	If VertexID Enable is ENABLED, this field specifies the 32-bit component location (within the 4-component VUE) where it is inserted. If VertexID Enable is DISABLED, this field is ignored.	
	Value	Name Description
	0	COMP_0 If enabled, VertexID is inserted in component 0 (.x)
	1	COMP_1 If enabled, VertexID is inserted in component 1 (.y)
	2	COMP_2 If enabled, VertexID is inserted in component 2 (.z)
3	COMP_3 If enabled, VertexID is inserted in component 3 (.w)	
12:6	Reserved	
	Format: MBZ	
5:0	VertexID Element Offset	
	Format: U6 Offset of 128-bit element	
	If VertexID Enable is ENABLED, this field specifies the VUE element offset of the 128-bit element where it is to be inserted. The VertexID Component Number specifies where in the specified element it is inserted. This is also the vertex element index. If VertexID Enable is DISABLED, this field is ignored.	
	Value	Name
[0,33]		



3DSTATE_VF_STATISTICS

3DSTATE_VF_STATISTICS					
Source:	RenderCS				
Length Bias:	1				
<p>The VF stage tracks two pipeline statistics, the number of vertices fetched and the number of objects generated. VF will increment the appropriate counter for each when statistics gathering is enabled by issuing the 3DSTATE_VF_STATISTICS command with the [Statistics Enable] bit set.</p>					
DWord	Bit	Description			
0	31:29	Command Type			
		Default Value:	3h GFXPIPE		
		Format:	Opcode		
	28:27	Command SubType			
		Format:	Opcode		
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td>GFXPIPE_SINGLE_DW [Default]</td> </tr> </tbody> </table>	Value	Name	1h
	Value	Name			
	1h	GFXPIPE_SINGLE_DW [Default]			
	26:24	3D Command Opcode			
		Default Value:	0h 3DSTATE_PIPELINED		
Format:		Opcode			
GFXPIPE[28:27 = 1h, 26:24 = 0h, 23:16 = 0Bh] (Pipelined, Single DWord)					
23:16	3D Command Sub Opcode				
	Default Value:	0Bh 3DSTATE_VF_STATISTICS			
	Format:	Opcode			
GFXPIPE[28:27 = 1h, 26:24 = 0h, 23:16 = 0Bh] (Pipelined, Single DWord)					
15:1	Reserved				
	Format:	MBZ			
0	Statistics Enable				
	Format:	Enable			
<p>If ENABLED, VF will increment the pipeline statistics counters IA_VERTICES_COUNT and IA_PRIMITIVES_COUNT for each vertex fetched and each object output, respectively, for 3DPRIMITIVE commands issued subsequently.</p> <p>If DISABLED, these counters will not be incremented for subsequent 3DPRIMITIVE commands.</p>					



3DSTATE_VF_TOPOLOGY

3DSTATE_VF_TOPOLOGY			
Source:	RenderCS		
Length Bias:	2		
This command specifies the VF stage's Topology state which can be used to override the Primitive Topology Type in subsequent 3DPRIMITIVE commands.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	4Bh 3DSTATE_VF_TOPOLOGY	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:6	Reserved	
		Format:	MBZ
	5:0	Primitive Topology Type	
Format:		3D_Prim_Topo_Type	
This field specifies the VF stage's Topology state.			



3DSTATE_VIEWPORT_STATE_POINTERS_CC

3DSTATE_VIEWPORT_STATE_POINTERS_CC			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_VIEWPORT_STATE_POINTERS_CC command is used to define the location of fixed functions' viewport state table.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	23h 3DSTATE_VIEWPORT_STATE_POINTERS_CC	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h DWORD_COUNT_n	
	Format:	=n	
1	31:5	CC Viewport Pointer	
		Format: DynamicStateOffset[31:5]CC_VIEWPORT*16 Specifies the 32-byte aligned address offset of the CC_VIEWPORT state. This offset is relative to the Dynamic State Base Address.	
1	4:0	Reserved	
		Format: MBZ	



3DSTATE_VIEWPORT_STATE_POINTERS_SF_CLIP

3DSTATE_VIEWPORT_STATE_POINTERS_SF_CLIP			
Source:	RenderCS		
Length Bias:	2		
The 3DSTATE_VIEWPORT_STATE_POINTERS_CLIP command is used to define the location of fixed functions' viewport state table.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	21h 3DSTATE_VIEWPORT_STATE_POINTERS_SF_CLIP	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h DWORD_COUNT_n	
	Format:	=n	
1	31:6	SF Clip Viewport Pointer	
		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	



3DSTATE_VS

3DSTATE_VS			
Source:	RenderCS, PositionCS		
Length Bias:	2		
Description			
This command specifies most of the state used by the Vertex Shader (VS) stage.			
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:		0h 3DSTATE_PIPELINED	
Format:		OpCode	
23:16	3D Command Sub Opcode		
	Default Value:	10h 3DSTATE_VS	
	Format:	OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	7h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1..2	63:6	Kernel Start Pointer	
		Format: InstructionBaseOffset[63:6]Kernel This field specifies the starting location of the kernel program run by threads spawned by the VS pipeline stage. It is specified as a 64-byte-granular offset from the Instruction Base Address. This field is ignored if VS Function Enable is DISABLED.	
	5:0	Reserved	
		Format: MBZ	
3	31	Single Vertex Dispatch	
		Format: U1 Enumerated Type When this bit is set, SIMD4x2 VS threads will only process a single vertex, otherwise SIMD4x2 threads will process either one or two vertices. This field is ignored if SIMD8 Dispatch Enable is	



3DSTATE_VS		
	set.	
	Value	Name Description
	0h	Multiple Dual vertex SIMD4x2 thread dispatches are allowed.
	1h	Single Single vertex SIMD4x2 thread dispatches are forced.
30	Vector Mask Enable	
	Format:	Enable
	Upon subsequent VS thread dispatches, this bit is loaded into the EU's Vector Mask Enable (VME, cr0.0[3]) thread state. Refer to EU documentation for the definition and use of VME state.	
	Value	Name Description
	0h	Dmask The EU will use the Dispatch Mask (supplied by the VS stage) for instruction execution.
	1h	Vmask The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.
	Programming Notes	
	Under normal conditions SW shall specify DMask, as the VS stage will provide a Dispatch Mask appropriate to SIMD4x2 or SIMD8 thread execution (as a function of SIMD8 Dispatch Enable). E.g., for SIMD4x2 thread execution, the VS stage will generate a Dispatch Mask that is equal to what the EU would use as the Vector Mask. For SIMD8 execution there is no known usage model for use of Vector Mask (as there is for PS shaders).	
29:27	Sampler Count	
	Format:	U3
	This field specifies (in multiples of 4) the number of sets of sampler state that will be prefetched for use by the VS kernel. While the prefetching of sampler state is optional and does not impact functionality, it may improve performance.	
	This field is ignored if the Function Enable state is set to DISABLED.	
	Value	Name Description
	0h	No Samplers no samplers used
	1h	1-4 Samplers between 1 and 4 samplers used
	2h	5-8 Samplers between 5 and 8 samplers used
	3h	9-12 Samplers between 9 and 12 samplers used
	4h	13-16 Samplers between 13 and 16 samplers used
26	Reserved	
	Format:	MBZ
25:18	Binding Table Entry Count	
	Format:	U8
	Description	
	Specifies how many binding table entries the kernel uses. Used only for prefetching of the	



3DSTATE_VS													
	<p>binding table entries and associated surface state. Note: 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. This field is ignored if VS Function Enable is DISABLED.</p> <p>When HW Generated Binding Table bit is enabled: This field indicates which cache lines (512bit units - 32 Binding Table Entry section) should be fetched. Each bit in this field corresponds to a cache line. Only the 1st 4 non-zero Binding Table entries of each 32 Binding Table entry section prefetched will have its surface state prefetched.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,255]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.</p>		Value	Name	[0,255]								
Value	Name												
[0,255]													
17	<p>Thread Dispatch Priority</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>U1 Enumerated Type</td> </tr> </table> <p>Specifies the priority of the thread for dispatch: This field is ignored if VS Function Enable is DISABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%; text-align: center;">Value</th> <th style="width: 25%; text-align: center;">Name</th> <th style="width: 50%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">Normal</td> <td style="text-align: center;">Normal Priority</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">High</td> <td style="text-align: center;">High Priority</td> </tr> </tbody> </table>		Format:	U1 Enumerated Type	Value	Name	Description	0h	Normal	Normal Priority	1h	High	High Priority
Format:	U1 Enumerated Type												
Value	Name	Description											
0h	Normal	Normal Priority											
1h	High	High Priority											
16	<p>Floating Point Mode</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>U1 Enumerated Type</td> </tr> </table> <p>Specifies the initial floating point mode used by the dispatched thread. This field is ignored if VS Function Enable is DISABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%; text-align: center;">Value</th> <th style="width: 25%; text-align: center;">Name</th> <th style="width: 50%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">IEEE-754</td> <td style="text-align: center;">Use IEEE-754 Rules</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">Alternate</td> <td style="text-align: center;">Use Alternate Rules</td> </tr> </tbody> </table>		Format:	U1 Enumerated Type	Value	Name	Description	0h	IEEE-754	Use IEEE-754 Rules	1h	Alternate	Use Alternate Rules
Format:	U1 Enumerated Type												
Value	Name	Description											
0h	IEEE-754	Use IEEE-754 Rules											
1h	Alternate	Use Alternate Rules											
15:14	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ									
Format:	MBZ												
13	<p>Illegal Opcode Exception Enable</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment. This field is ignored if VS Function Enable is DISABLED.</p>		Format:	Enable									
Format:	Enable												



3DSTATE_VS										
	<p>12 Accesses UAV</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This field must be set when VS has a UAV access.</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table> <p>This field must not be set when VS Function Enable is disabled.</p>	Format:	Enable	Programming Notes						
	Format:	Enable								
	Programming Notes									
	<p>11:8 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
<p>7 Software Exception Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit gets loaded into EU CR0.1[13] (note the bit # difference). See Exceptions and ISA Execution Environment. This field is ignored if VS Function Enable is DISABLED.</p>	Format:	Enable								
Format:	Enable									
<p>6:0 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ									
4..5	<p>63:10 Scratch Space Base Pointer</p> <table border="1"> <tr> <td>Format:</td> <td>GeneralStateOffset[63:10]ScratchSpace</td> </tr> </table> <p>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 VS Function Enable is DISABLED. In 64b OS all pointers need to be seen by SW as 48b. HW does not support a Scratch Space Base Pointer larger than 32b, therefore SW must ensure Bits<63:32> are set to 0's.</p>	Format:	GeneralStateOffset[63:10]ScratchSpace							
	Format:	GeneralStateOffset[63:10]ScratchSpace								
	<p>9:4 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
<p>3:0 Per-Thread Scratch Space</p> <table border="1"> <tr> <td>Format:</td> <td>U4 power of 2 Bytes over 1K Bytes</td> </tr> </table> <p>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 VS Function Enable is DISABLED.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td></td> <td>Indicating [1K Bytes, 2M Bytes]</td> </tr> </tbody> </table> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table>	Format:	U4 power of 2 Bytes over 1K Bytes	Value	Name	Description	[0,11]		Indicating [1K Bytes, 2M Bytes]	Programming Notes	
Format:	U4 power of 2 Bytes over 1K Bytes									
Value	Name	Description								
[0,11]		Indicating [1K Bytes, 2M Bytes]								
Programming Notes										



3DSTATE_VS												
		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.										
6	31:25	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">MBZ</td></tr></table>		MBZ								
		MBZ										
	24:20	Dispatch GRF Start Register For URB Data Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">U5</td></tr></table> Specifies the starting GRF number for the URB portion (URB constants and vertices) of the thread payload. This field is ignored if VS Function Enable is DISABLED. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td></td> <td>indicating GRF [R0, R31]</td> </tr> </tbody> </table>		U5	Value	Name	Description	[0,31]		indicating GRF [R0, R31]		
		U5										
	Value	Name	Description									
	[0,31]		indicating GRF [R0, R31]									
	19:17	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">MBZ</td></tr></table>		MBZ								
		MBZ										
16:11	Vertex URB Entry Read Length Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">U6</td></tr></table> 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. For SIMD4x2 dispatch, each vertex element requires one GRF of payload data, therefore the number of GRFs with vertex data will be double the value programmed in this field. For SIMD8 dispatch, each vertex element requires 4 GRFs of payload data, therefore the number of GRFs with vertex data will be 8 times the value programmed in this field. The EU limit of 128 GRFs imposes a maximum limit of 30 elements per vertex pushed into the payload, though the practical limit may be lower. If input vertices exceed the practical limit, software must decide between resorting to pulling elements during thread execution or dropping back to SIMD4x2 dispatch. Note that the VUE is used for both input and output, so when using the pull-model software must ensure inputs are not overwritten before last use. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>[1,63]</td> <td></td> <td>if SIMD8 dispatch disabled</td> </tr> <tr> <td>[0,15]</td> <td></td> <td>if SIMD8 dispatch enabled</td> </tr> </tbody> </table>		U6	Value	Name	Description	[1,63]		if SIMD8 dispatch disabled	[0,15]		if SIMD8 dispatch enabled
	U6											
Value	Name	Description										
[1,63]		if SIMD8 dispatch disabled										
[0,15]		if SIMD8 dispatch enabled										
10	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">MBZ</td></tr></table>		MBZ									
	MBZ											
9:4	Vertex URB Entry Read Offset Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">U6</td></tr></table> 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. This field is ignored if VS Function Enable is DISABLED. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td style="height: 20px;"></td> <td></td> </tr> </tbody> </table>		U6	Value	Name							
	U6											
Value	Name											



3DSTATE_VS								
		[0,63]						
	3:0	Reserved Format: MBZ						
7	31:23	Maximum Number of Threads Format: U9-1 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 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 VS Function Enable is DISABLED.						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,335]</td> <td></td> <td>indicating thread count of [1,336]</td> </tr> </tbody> </table>	Value	Name	Description	[0,335]		indicating thread count of [1,336]
	Value	Name	Description					
	[0,335]		indicating thread count of [1,336]					
	22	Reserved Format: MBZ						
	21:13	Reserved Format: MBZ						
	12:11	Reserved Format: MBZ						
	10	Statistics Enable Format: Enable If ENABLED, the VS stage will perform statistics gathering. See the Statistics Gathering subsection. If DISABLED, statistics information associated with the VS stage will be left unchanged.						
	9	Reserved Format: MBZ						
	8:3	Reserved Format: MBZ						
2	SIMD8 Dispatch Enable Format: Enable This field determines how VS threads are dispatched and how the thread payloads are generated. The setting of this field must agree with how the VS kernel was compiled. If ENABLED, SIMD8 VS thread dispatches are performed. The Single Vertex Dispatch field is ignored. If DISABLED, SIMD4x2 thread dispatches are performed. The Single Vertex Dispatch field can be used to force single-vertex dispatches.							



3DSTATE_VS						
1	<p>Vertex Cache Disable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Disable</td> </tr> </table> <p>This bit controls the operation of the Vertex Cache. This field is always used. If the Vertex Cache is DISABLED and the VS Function is ENABLED, the Vertex Cache is not used and all incoming vertices will be passed to VS threads. If the Vertex Cache is ENABLED and the VS Function is ENABLED, only incoming vertices that do not hit in the Vertex Cache will be passed to VS threads. If the Vertex Cache is ENABLED and the VS Function is DISABLED, input vertices that miss in the Vertex Cache will be assembled and written to the URB (by the VF stage), and subsequently passed through the VS stage unmodified (i.e, no VS threads are spawned). The Vertex Cache is invalidated whenever the Vertex Cache becomes DISABLED, whenever the VS Function Enable toggles, between 3DPRIMITIVE commands and between instances within a 3DPRIMITIVE command.</p>	Format:	Disable			
	Format:	Disable				
<p>Function Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit determines whether or not the VS stage spawns VS threads, which comprises the bulk of the VS stage functionality. 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 are sent down the pipeline unmodified. The Vertex Cache (if enabled) is still available.</p>	Format:	Enable				
Format:	Enable					
8	<p>31:28 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ				
	<p>27 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
<p>26:21 Vertex URB Entry Output Read Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U6</td> </tr> </table> <p>Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB by the Setup Back-End (SBE) function. The offset programmed will specify the start of Attribute 0 to be passed in subsequent Pixel Shader thread payloads. Refer to the Attribute Interpolator Setup documentation.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,63]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>As the vertex header data located at the start of the Vertex URB entry is typically only used by 3D pipeline FFs (i.e., Clipper, Setup FrontEnd) and not required as interpolated attributes in Pixel Shader threads, it is expected that SW will program this Start Offset skip over the vertex header. This offset value is ignored if SBE's Number of SF Attributes state is programmed to 0 (i.e., no</p>	Format:	U6	Value	Name	[0,63]	
Format:	U6					
Value	Name					
[0,63]						



3DSTATE_VS							
	attributes are defined beyond the position read from the Vertex Header)						
20:16	<p>Vertex URB Entry Output Length</p> <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Specifies the amount of Vertex Attribute URB data to be read by the Setup Back-End function for each Vertex URB entry, in 256-bit units. The attribute data will be read starting at the offset specified by the Vertex URB Entry Output Read Offset state.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[1,16]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This length value is ignored if SBE's Number of SF Attributes state is programmed to 0 (i.e., no attributes are defined beyond the position read from the Vertex Header).</p>	Format:	U5	Value	Name	[1,16]	
Format:	U5						
Value	Name						
[1,16]							
15:8	<p>User Clip Distance Clip Test Enable Bitmask</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>This 8 bit mask field selects which of the 8 Clip Distance Values (if any) are to be included in the Clip stage's trivial reject / trivial accept / must clip determination function. The ClipDistance Values (if present) are located in DW8-15 of the VUE Vertex Header located at the beginning of VUE URB entries. Bit 0 of this field corresponds to Clip Distance Value 0.</p>	Format:	U8				
Format:	U8						
7:0	<p>User Clip Distance Cull Test Enable Bitmask</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>This 8 bit mask field selects which of the 8 Clip Distance Values (if any) are to be included in the Clip stage's trivial reject / trivial accept determination function. Note that must clip determination is not included in this function. The ClipDistance Values (if present) are located in DW8-15 of the VUE Vertex Header located at the beginning of VUE URB entries. Bit 0 of this field corresponds to Clip Distance Value 0.</p>	Format:	U8				
Format:	U8						



3DSTATE_WM_CHROMAKEY

3DSTATE_WM_CHROMAKEY		
Source:		RenderCS
Length 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: 0h 3DSTATE_PIPELINED Format: OpCode
	23:16	3D Command Sub Opcode
		Default Value: 4Ch 3DSTATE_WM_CHROMAKEY Format: OpCode
	15:8	Reserved
		Format: MBZ
7:0	Dword Length	
	Default Value: 0h Excludes Dword (0,1) Format: =n	
	Total Length - 2	
1	31	ChromaKey Kill Enable
		Format: Enable If ENABLED, indicates that at least one of the attached samplers has ChromaKeyKill enabled.
1	30:0	Reserved
		Format: MBZ



3DSTATE_WM_DEPTH_STENCIL

3DSTATE_WM_DEPTH_STENCIL			
Source:		RenderCS	
Length Bias:		2	
This command replaces the indirect state DEPTH_STENCIL_STATE with an inline state command.			
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		4Eh 3DSTATE_WM_DEPTH_STENCIL	
Format:		OpCode	
15:13	Reserved		
	Format:	MBZ	
12:8	Reserved		
	Format:	MBZ	
7:0	Dword Length		
	Format:	=n	
	Total Length - 2		
	Value	Name	
	02h	Excludes Dword (0,1) [Default]	
1	31:29	Stencil Fail Op	
		Format:	3D_Stencil_Operation
		This field specifies the operation to perform on the Stencil Buffer when the (front face) stencil test fails.	
	Programming Notes		
	if all three stencil ops (Stencil Fail, Stencil Pass Depth Fail, and Stencil Pass Depth Pass) are KEEP, ZERO, or REPLACE, the stencil buffer is not read.		
	28:26	Stencil Pass Depth Fail Op	



3DSTATE_WM_DEPTH_STENCIL		
	Format:	3D_Stencil_Operation
	This field specifies the operation to perform on the Stencil Buffer when the (front face) stencil test passes but the depth pass fails.	
25:23	Stencil Pass Depth Pass Op	
	Format:	3D_Stencil_Operation
	This field specifies the operation to perform on the Stencil Buffer when the (front face) stencil test passes but the depth test passes.	
22:20	Backface Stencil Test Function	
	Format:	3D_Compare_Function
19:17	Backface Stencil Fail Op	
	Format:	3D_Stencil_Operation
16:14	Backface Stencil Pass Depth Fail Op	
	Format:	3D_Stencil_Operation
	This field specifies the operation to perform on the Stencil Buffer when the stencil test passes but the depth pass fails.	
13:11	Backface Stencil Pass Depth Pass Op	
	Format:	3D_Stencil_Operation
	This field specifies the operation to perform on the Stencil Buffer when the stencil test passes and the depth pass passes (or is disabled).	
10:8	Stencil Test Function	
	Format:	3D_Compare_Function
	This field specifies the comparison function used in the (front face) StencilTest function.	
7:5	Depth Test Function	
	Format:	3D_Compare_Function
	Specifies the comparison function used in DepthTest function.	
	Programming Notes	
	If the Depth Test Function is ALWAYS or NEVER, the depth buffer is not read.	
4	Double Sided Stencil Enable	
	Format:	Enable
	Enable doubled sided stencil operations.	
	Value	Name Description
	0h	False Double Sided Stencil Disabled
	1h	True Double Sided Stencil Enabled



3DSTATE_WM_DEPTH_STENCIL													
	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2"> <ul style="list-style-type: none"> Back-facing primitives have a vertex winding order opposite to the currently selected Front Winding state. Culling of primitives is not affected by the double sided stencil state Back-facing primitives will be rendered, honoring all current device state, as though it were a front-facing primitive with no implicitly overloaded state. </td> </tr> </table>	Programming Notes		<ul style="list-style-type: none"> Back-facing primitives have a vertex winding order opposite to the currently selected Front Winding state. Culling of primitives is not affected by the double sided stencil state Back-facing primitives will be rendered, honoring all current device state, as though it were a front-facing primitive with no implicitly overloaded state. 									
Programming Notes													
<ul style="list-style-type: none"> Back-facing primitives have a vertex winding order opposite to the currently selected Front Winding state. Culling of primitives is not affected by the double sided stencil state Back-facing primitives will be rendered, honoring all current device state, as though it were a front-facing primitive with no implicitly overloaded state. 													
	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">3</td> <td> Stencil Test Enable Format: Enable Enables StencilTest function of the Pixel Processing pipeline. <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If any of the render targets are YUV format, this field must be disabled.</td> </tr> </table> </td> </tr> </table>	3	Stencil Test Enable Format: Enable Enables StencilTest function of the Pixel Processing pipeline. <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If any of the render targets are YUV format, this field must be disabled.</td> </tr> </table>	Programming Notes		If any of the render targets are YUV format, this field must be disabled.							
3	Stencil Test Enable Format: Enable Enables StencilTest function of the Pixel Processing pipeline. <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If any of the render targets are YUV format, this field must be disabled.</td> </tr> </table>	Programming Notes		If any of the render targets are YUV format, this field must be disabled.									
Programming Notes													
If any of the render targets are YUV format, this field must be disabled.													
	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">2</td> <td> Stencil Buffer Write Enable Format: Enable Enables writes to the Stencil Buffer. <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If this field is enabled, Stencil Test Enable must also be enabled.</td> </tr> </table> </td> </tr> </table>	2	Stencil Buffer Write Enable Format: Enable Enables writes to the Stencil Buffer. <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If this field is enabled, Stencil Test Enable must also be enabled.</td> </tr> </table>	Programming Notes		If this field is enabled, Stencil Test Enable must also be enabled.							
2	Stencil Buffer Write Enable Format: Enable Enables writes to the Stencil Buffer. <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If this field is enabled, Stencil Test Enable must also be enabled.</td> </tr> </table>	Programming Notes		If this field is enabled, Stencil Test Enable must also be enabled.									
Programming Notes													
If this field is enabled, Stencil Test Enable must also be enabled.													
	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">1</td> <td> Depth Test Enable Format: Enable Enables the DepthTest function of the Pixel Processing pipeline. <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">Disable</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">Enable</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If any of the render targets are YUV format, this field must be disabled.</td> </tr> </table> </td> </tr> </table>	1	Depth Test Enable Format: Enable Enables the DepthTest function of the Pixel Processing pipeline. <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">Disable</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">Enable</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If any of the render targets are YUV format, this field must be disabled.</td> </tr> </table>	Value	Name	0h	Disable	1h	Enable	Programming Notes		If any of the render targets are YUV format, this field must be disabled.	
1	Depth Test Enable Format: Enable Enables the DepthTest function of the Pixel Processing pipeline. <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">Disable</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">Enable</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If any of the render targets are YUV format, this field must be disabled.</td> </tr> </table>	Value	Name	0h	Disable	1h	Enable	Programming Notes		If any of the render targets are YUV format, this field must be disabled.			
Value	Name												
0h	Disable												
1h	Enable												
Programming Notes													
If any of the render targets are YUV format, this field must be disabled.													
	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">0</td> <td> Depth Buffer Write Enable Format: Enable Enables writes to the Depth Buffer. <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">A Depth Buffer must be defined before enabling writes to it, or operation is UNDEFINED.</td> </tr> <tr> <td colspan="2">This bit must not be set when WM_INT::RT Independent Rasterization Enable is true.</td> </tr> </table> </td> </tr> </table>	0	Depth Buffer Write Enable Format: Enable Enables writes to the Depth Buffer. <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">A Depth Buffer must be defined before enabling writes to it, or operation is UNDEFINED.</td> </tr> <tr> <td colspan="2">This bit must not be set when WM_INT::RT Independent Rasterization Enable is true.</td> </tr> </table>	Programming Notes		A Depth Buffer must be defined before enabling writes to it, or operation is UNDEFINED.		This bit must not be set when WM_INT::RT Independent Rasterization Enable is true.					
0	Depth Buffer Write Enable Format: Enable Enables writes to the Depth Buffer. <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">A Depth Buffer must be defined before enabling writes to it, or operation is UNDEFINED.</td> </tr> <tr> <td colspan="2">This bit must not be set when WM_INT::RT Independent Rasterization Enable is true.</td> </tr> </table>	Programming Notes		A Depth Buffer must be defined before enabling writes to it, or operation is UNDEFINED.		This bit must not be set when WM_INT::RT Independent Rasterization Enable is true.							
Programming Notes													
A Depth Buffer must be defined before enabling writes to it, or operation is UNDEFINED.													
This bit must not be set when WM_INT::RT Independent Rasterization Enable is true.													
2	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">31:24</td> <td> Stencil Test Mask Format: U8 This field specifies a bit mask applied to stencil test values. Both the stencil reference value and value read from the stencil buffer will be logically ANDed with this mask before the stencil comparison test is performed. </td> </tr> </table>	31:24	Stencil Test Mask Format: U8 This field specifies a bit mask applied to stencil test values. Both the stencil reference value and value read from the stencil buffer will be logically ANDed with this mask before the stencil comparison test is performed.										
31:24	Stencil Test Mask Format: U8 This field specifies a bit mask applied to stencil test values. Both the stencil reference value and value read from the stencil buffer will be logically ANDed with this mask before the stencil comparison test is performed.												



3DSTATE_WM_DEPTH_STENCIL				
	23:16	Stencil Write Mask Format: <table border="1" style="display: inline-table;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U8</td></tr></table> This field specifies a bit mask applied to stencil buffer writes. Only those stencil buffer bits corresponding to bits set in this mask will be modified.		U8
		U8		
	15:8	Backface Stencil Test Mask Format: <table border="1" style="display: inline-table;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U8</td></tr></table> This field specifies a bit mask applied to backface stencil test values. Both the stencil reference value and value read from the stencil buffer will be logically ANDed with this mask before the stencil comparison test is performed.		U8
	U8			
7:0	Backface Stencil Write Mask Format: <table border="1" style="display: inline-table;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U8</td></tr></table> This field specifies a bit mask applied to backface stencil buffer writes. Only those stencil buffer bits corresponding to bits set in this mask will be modified.		U8	
	U8			
3	31:16	Reserved Format: <table border="1" style="display: inline-table;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">MBZ</td></tr></table>		MBZ
		MBZ		
	15:8	Stencil Reference Value Format: <table border="1" style="display: inline-table;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U8</td></tr></table> This field specifies the stencil reference value to compare against in the (front face) StencilTest function.		U8
	U8			
7:0	Backface Stencil Reference Value Format: <table border="1" style="display: inline-table;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U8</td></tr></table> This field specifies the stencil reference value to compare against in the StencilTest function.		U8	
	U8			



3DSTATE_WM

3DSTATE_WM			
Source:		RenderCS	
Length 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:	0h 3DSTATE_PIPELINED
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		14h 3DSTATE_WM	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31	Statistics Enable	
		Format:	Enable
		If ENABLED, the Windower and pixel pipeline will engage in statistics gathering. If DISABLED, statistics information associated with this FF stage will be left unchanged. See Statistics Gathering.	
Programming Notes			
This bit must be disabled if any of these bits is set: 3DSTATE_WM::Legacy Depth Buffer Clear, 3DSTATE_WM::Legacy Hierarchical Depth Buffer Resolve Enable or 3DSTATE_WM::Legacy Depth Buffer Resolve Enable.			
	30	Legacy Depth Buffer Clear Enable	
		Format:	Enable
		When set, the depth buffer is initialized as a side-effect of rendering pixels.	

**3DSTATE_WM**

3DSTATE_WM											
	<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> <p>If this field is enabled,</p> <ol style="list-style-type: none"> the Depth Test Enable field in DEPTH_STENCIL_STATE must be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. 3DSTATE_DEPTH_BUFFER::Stencil Write Enable must be set if 3DSTATE_STENCIL_BUFFER::Stencil buffer enable is set. Additionally the following must be set to the correct values. <ol style="list-style-type: none"> 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 DEPTH_STENCIL_STATE::Back Face Stencil Test Mask must be 0xFF <p>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.</p> </td> </tr> </tbody> </table>	Programming Notes		<p>If this field is enabled,</p> <ol style="list-style-type: none"> the Depth Test Enable field in DEPTH_STENCIL_STATE must be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. 3DSTATE_DEPTH_BUFFER::Stencil Write Enable must be set if 3DSTATE_STENCIL_BUFFER::Stencil buffer enable is set. Additionally the following must be set to the correct values. <ol style="list-style-type: none"> 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 DEPTH_STENCIL_STATE::Back Face Stencil Test Mask must be 0xFF <p>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.</p>							
Programming Notes											
<p>If this field is enabled,</p> <ol style="list-style-type: none"> the Depth Test Enable field in DEPTH_STENCIL_STATE must be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. 3DSTATE_DEPTH_BUFFER::Stencil Write Enable must be set if 3DSTATE_STENCIL_BUFFER::Stencil buffer enable is set. Additionally the following must be set to the correct values. <ol style="list-style-type: none"> 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 DEPTH_STENCIL_STATE::Back Face Stencil Test Mask must be 0xFF <p>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.</p>											
29	<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2">Reserved</th> </tr> </thead> <tbody> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </tbody> </table>	Reserved		Format:	MBZ						
Reserved											
Format:	MBZ										
28	<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2">Legacy Depth Buffer Resolve Enable</th> </tr> </thead> <tbody> <tr> <td>Format:</td> <td style="text-align: center;">Enable</td> </tr> <tr> <td colspan="2"> <p>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.</p> </td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2"> <p>If this field is enabled,</p> <ol style="list-style-type: none"> the Legacy Depth Buffer Clear and Legacy Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. <p>Refer to section 11.5.4.2 "Depth Buffer Resolve" for additional restrictions when this field is enabled.</p> <p>If Hierarchical Depth Buffer Enable is disabled, enabling this field will have no effect.</p> </td> </tr> </tbody> </table>	Legacy Depth Buffer Resolve Enable		Format:	Enable	<p>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.</p>		Programming Notes		<p>If this field is enabled,</p> <ol style="list-style-type: none"> the Legacy Depth Buffer Clear and Legacy Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. <p>Refer to section 11.5.4.2 "Depth Buffer Resolve" for additional restrictions when this field is enabled.</p> <p>If Hierarchical Depth Buffer Enable is disabled, enabling this field will have no effect.</p>	
Legacy Depth Buffer Resolve Enable											
Format:	Enable										
<p>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.</p>											
Programming Notes											
<p>If this field is enabled,</p> <ol style="list-style-type: none"> the Legacy Depth Buffer Clear and Legacy Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. <p>Refer to section 11.5.4.2 "Depth Buffer Resolve" for additional restrictions when this field is enabled.</p> <p>If Hierarchical Depth Buffer Enable is disabled, enabling this field will have no effect.</p>											
27	<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2">Legacy Hierarchical Depth Buffer Resolve Enable</th> </tr> </thead> <tbody> <tr> <td>Format:</td> <td style="text-align: center;">Enable</td> </tr> <tr> <td colspan="2"> <p>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.</p> </td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2"> <p>If this field is enabled,</p> <ol style="list-style-type: none"> the Legacy Depth Buffer Clear and Legacy Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. </td> </tr> </tbody> </table>	Legacy Hierarchical Depth Buffer Resolve Enable		Format:	Enable	<p>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.</p>		Programming Notes		<p>If this field is enabled,</p> <ol style="list-style-type: none"> the Legacy Depth Buffer Clear and Legacy Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. 	
Legacy Hierarchical Depth Buffer Resolve Enable											
Format:	Enable										
<p>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.</p>											
Programming Notes											
<p>If this field is enabled,</p> <ol style="list-style-type: none"> the Legacy Depth Buffer Clear and Legacy Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. 											



3DSTATE_WM																					
	<p>Refer to section 11.5.4.3 "Hierarchical Depth Buffer Resolve" for additional restrictions when this field is enabled.</p> <p>If Hierarchical Depth Buffer Enable is disabled, enabling this field will have no effect.</p> <p>Performance Note: expect the hierarchical depth buffer's impact on performance to be reduced for some period of time after this operation is performed, as the hierarchical depth buffer is initialized to a state that makes it ineffective. Further rendering will tend to bring the hierarchical depth buffer back to a more effective state.</p>																				
26	<p>Legacy Diamond Line Rasterization</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the Windower will rasterize zero width lines using the DX9 rasterization rules. If DISABLED, the Windower will rasterize zero width lines using the DX10 rasterization rules (see Strips Fans chapter).</p>		Format:	Enable																	
Format:	Enable																				
25:23	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ																	
Format:	MBZ																				
22:21	<p>Early Depth/Stencil Control</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>U2 Enumerated Type</td> </tr> </table> <p>This field specifies the behavior of early depth/stencil test.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NORMAL</td> <td>Depth/Stencil Test/Write behaves as if it happens post-shader, however the pixel shader is not necessarily executed if the pixel fails depth or stencil test (this is the legacy behavior)</td> </tr> <tr> <td>1h</td> <td>PSEXEC</td> <td>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)</td> </tr> <tr> <td>2h</td> <td>PREPS</td> <td>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.</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 100%;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>The Early Depth/Stencil Control field cannot be set to PREPS (value = 2h) if ForceKillpix = ForceON or Forced Thread Dispatch = ForceON</td> </tr> </tbody> </table>		Format:	U2 Enumerated Type	Value	Name	Description	0h	NORMAL	Depth/Stencil Test/Write behaves as if it happens post-shader, however the pixel shader is not necessarily executed if the pixel fails depth or stencil test (this is the legacy behavior)	1h	PSEXEC	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)	2h	PREPS	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	The Early Depth/Stencil Control field cannot be set to PREPS (value = 2h) if ForceKillpix = ForceON or Forced Thread Dispatch = ForceON
Format:	U2 Enumerated Type																				
Value	Name	Description																			
0h	NORMAL	Depth/Stencil Test/Write behaves as if it happens post-shader, however the pixel shader is not necessarily executed if the pixel fails depth or stencil test (this is the legacy behavior)																			
1h	PSEXEC	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)																			
2h	PREPS	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																					
The Early Depth/Stencil Control field cannot be set to PREPS (value = 2h) if ForceKillpix = ForceON or Forced Thread Dispatch = ForceON																					
20:19	<p>Force Thread Dispatch Enable</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Normal</td> <td>WM_INT::ThreadDispatchEnable is computed normally</td> </tr> <tr> <td>1h</td> <td>ForceOff</td> <td>Forces WM_INT::ThreadDispatchEnable Off</td> </tr> <tr> <td>2h</td> <td>ForceON</td> <td>Forces WM_INT::ThreadDispatchEnable On</td> </tr> </tbody> </table>		Value	Name	Description	0h	Normal	WM_INT::ThreadDispatchEnable is computed normally	1h	ForceOff	Forces WM_INT::ThreadDispatchEnable Off	2h	ForceON	Forces WM_INT::ThreadDispatchEnable On							
Value	Name	Description																			
0h	Normal	WM_INT::ThreadDispatchEnable is computed normally																			
1h	ForceOff	Forces WM_INT::ThreadDispatchEnable Off																			
2h	ForceON	Forces WM_INT::ThreadDispatchEnable On																			



3DSTATE_WM		
	3h	Reserved
Programming Notes		
This should must always be set to Normal. This field should not be tested for functional validation		
18:17	Position ZW Interpolation Mode	
	Format:	U2 Enumerated Type
This field elects "interpolation mode" associated with the Position Z (source depth) and W coordinates passed in the PS payload when the PS requires Position as input. This field does not determine whether these coordinates are actually included in the payload (see Pixel Shader Requires Depth, Pixel Shader 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		
WM_INT::RT Independent Rasterization Enable must be disabled in order to select INTERP_SAMPLE.		
MSDISPMODE_PERSAMPLE is required in order to select INTERP_SAMPLE.		
16:11	Barycentric Interpolation Mode	
	Format:	Enable[6]
Controls which barycentric interpolation terms must be passed into the pixel shader kernel. Bit 0: Perspective Pixel Location barycentric is required Bit 1: Perspective Centroid barycentric is required Bit 2: Perspective Sample barycentric is required Bit 3: Non-perspective Pixel Location barycentric is required Bit 4: Non-perspective Centroid barycentric is required Bit 5: Non-perspective Sample 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.		
10	Reserved	
	Format:	MBZ



3DSTATE_WM			
9:8	Line End Cap Antialiasing Region Width		
	Format:	U2	
	This field specifies the distances over which the coverage of anti-aliased line end caps are computed.		
	Value	Name	Description
	0h	0.5 pixels	0.5 pixels
	1h	1.0 pixels	1.0 pixels
	2h	2.0 pixels	2.0 pixels
	3h	4.0 pixels	4.0 pixels
	7:6	Line Antialiasing Region Width	
		Format:	U2
This field specifies the distance over which the anti-aliased line coverage is computed.			
Value		Name	Description
0h		0.5 pixels	0.5 pixels
1h		1.0 pixels	1.0 pixels
2h		2.0 pixels	2.0 pixels
3h		4.0 pixels	4.0 pixels
5		Reserved	
Format:		MBZ	
4	Polygon Stipple Enable		
Format:	Enable		
Enables the Polygon Stipple function.			
3	Line Stipple Enable		
Format:	Enable		
Enables the Line Stipple function.			
2	Point Rasterization Rule		
Format:	3D_RasterizationRule		
This field specifies the rasterization rules to be applied whenever the edges of a point primitive fall exactly on a pixel sampling point.			
Value	Name	Description	
0h	RASTRULE_UPPER_LEFT	To match "normal" upper left rules for surface primitives	
1h	RASTRULE_UPPER_RIGHT	To match OpenGL point rasterization rules (round to + infinity, where this is the upper right direction wrt OpenGL screen origin of lower left).	
1:0	Force Kill Pixel Enable		
Value	Name	Description	



3DSTATE_WM		
0h	Normal	WM_INT:: Pixel Shader Kill Pixel is computed normally
1h	ForceOff	Forces WM_INT:: Pixel Shader Kill Pixel Off
2h	ForceON	Forces WM_INT:: Pixel Shader Kill Pixel On
3h	Reserved	
Programming Notes		
This should must always be set to Normal. This field should not be tested for functional validation		



3DSTATE_WM_HZ_OP

3DSTATE_WM_HZ_OP			
Source:	RenderCS		
Length Bias:	2		
This command provides for clearing Z and/or stencil or resolving either HZ buffer or Z buffer.			
Programming Notes			
As this command generates an implicit rectangle, SW must make sure any MMIO register writes following WM_HZ_OP must be preceded by PIPE_CONTROL with Command Streamer Stall Enable bit set.			
3DSTATE_MULTISAMPLE packet must be used prior to this packet to change the Number of Multisamples. This packet must not be used to change Number of Multisamples in a rendering sequence.			
3DSTATE_RASTER if used must be programmed prior to using this packet.			
This command does NOT support predication from the use of the MI_PREDICATE register. To predicate depth clears and resolves on you must fall back to using the 3D_PRIMITIVE or GPGPU_WALKER commands.			
Clear_Value_Test_Enable	Force_Slow_Clear_Enable	Clear_Value_Write_Enable	
0	0	0	Fast Clear; No Clear_value_address update
0	0	1	Fast Clear; Clear_value_address updated
0	1	0	Slow Clear; No Clear_value_address update
0	1	1	Illegal
1	0	0	Conditional Fast Clear; No Clear_value_address update
1	1	0	Illegal
1	X	1	Illegal
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:	0h 3DSTATE_PIPELINED
		Format:	OpCode



3DSTATE_WM_HZ_OP													
	<table border="1"> <tr> <td style="text-align: center;">23:16</td> <td>3D Command Sub Opcode</td> </tr> <tr> <td>Default Value:</td> <td>52h 3DSTATE_WM_HZ_OP</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	23:16	3D Command Sub Opcode	Default Value:	52h 3DSTATE_WM_HZ_OP	Format:	OpCode						
	23:16	3D Command Sub Opcode											
	Default Value:	52h 3DSTATE_WM_HZ_OP											
Format:	OpCode												
<table border="1"> <tr> <td style="text-align: center;">15:8</td> <td>Reserved</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	15:8	Reserved	Format:	MBZ									
15:8	Reserved												
Format:	MBZ												
<table border="1"> <tr> <td style="text-align: center;">7:0</td> <td>Dword Length</td> </tr> <tr> <td>Format:</td> <td>=n</td> </tr> <tr> <td colspan="2">Total Length - 2</td> </tr> <tr> <td style="text-align: center;">Value</td> <td style="text-align: center;">Name</td> </tr> <tr> <td>03h</td> <td>Excludes Dword (0,1) [Default]</td> </tr> </table>	7:0	Dword Length	Format:	=n	Total Length - 2		Value	Name	03h	Excludes Dword (0,1) [Default]			
7:0	Dword Length												
Format:	=n												
Total Length - 2													
Value	Name												
03h	Excludes Dword (0,1) [Default]												
1	<table border="1"> <tr> <td style="text-align: center;">31</td> <td>Stencil Buffer Clear Enable</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> <tr> <td colspan="2">When set, the stencil buffer is initialized.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">If this field is enabled,</td> </tr> <tr> <td colspan="2"> <ol style="list-style-type: none"> the Depth Buffer Resolve Enable and Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Stencil Write Enable must be set if 3DSTATE_STENCIL_BUFFER::Stencil buffer enable is set. </td> </tr> </table>	31	Stencil Buffer Clear Enable	Format:	Enable	When set, the stencil buffer is initialized.		Programming Notes		If this field is enabled,		<ol style="list-style-type: none"> the Depth Buffer Resolve Enable and Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Stencil Write Enable must be set if 3DSTATE_STENCIL_BUFFER::Stencil buffer enable is set. 	
31	Stencil Buffer Clear Enable												
Format:	Enable												
When set, the stencil buffer is initialized.													
Programming Notes													
If this field is enabled,													
<ol style="list-style-type: none"> the Depth Buffer Resolve Enable and Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Stencil Write Enable must be set if 3DSTATE_STENCIL_BUFFER::Stencil buffer enable is set. 													
	<table border="1"> <tr> <td style="text-align: center;">30</td> <td>Depth Buffer Clear Enable</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> <tr> <td colspan="2">When set, the depth buffer is initialized.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">If this field is enabled,</td> </tr> <tr> <td colspan="2"> <ol style="list-style-type: none"> the Depth Buffer Resolve Enable and Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. </td> </tr> </table>	30	Depth Buffer Clear Enable	Format:	Enable	When set, the depth buffer is initialized.		Programming Notes		If this field is enabled,		<ol style="list-style-type: none"> the Depth Buffer Resolve Enable and Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. 	
30	Depth Buffer Clear Enable												
Format:	Enable												
When set, the depth buffer is initialized.													
Programming Notes													
If this field is enabled,													
<ol style="list-style-type: none"> the Depth Buffer Resolve Enable and Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. 													
	<table border="1"> <tr> <td style="text-align: center;">29</td> <td>Scissor Rectangle Enable</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> <tr> <td colspan="2">Enables operation of Scissor Rectangle.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">In order get the functionality right if this bit is disabled, driver must clip the clear rectangle to scissor rectangle if scissor test is</td> </tr> </table>	29	Scissor Rectangle Enable	Format:	Enable	Enables operation of Scissor Rectangle.		Programming Notes		In order get the functionality right if this bit is disabled, driver must clip the clear rectangle to scissor rectangle if scissor test is			
29	Scissor Rectangle Enable												
Format:	Enable												
Enables operation of Scissor Rectangle.													
Programming Notes													
In order get the functionality right if this bit is disabled, driver must clip the clear rectangle to scissor rectangle if scissor test is													



3DSTATE_WM_HZ_OP			
	<p>enabled before clearing.</p> <p style="text-align: center;">Workaround</p> <p>Workaround: Due to a Hardware issue this bit must not be set.</p>		
28	<p>Depth Buffer Resolve Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>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.</p> <p style="text-align: center;">Programming Notes</p> <p>If this field is enabled,</p> <ol style="list-style-type: none"> 1. the Depth Buffer Clear and Hierarchical Depth Buffer Resolve Enable fields must both be disabled. 2. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. <p>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.</p>	Format:	Enable
Format:	Enable		
27	<p>Hierarchical Depth Buffer Resolve Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>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.</p> <p style="text-align: center;">Programming Notes</p> <p>If this field is enabled,</p> <ol style="list-style-type: none"> 1. the Depth Buffer Clear and Depth Buffer Resolve Enable fields must both be disabled. 2. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set. <p>Refer to section 11.5.4.3 "Hierarchical 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.</p> <p>Performance Note: expect the hierarchical depth buffer's impact on performance to be reduced for some period of time after this operation is performed, as the hierarchical depth buffer is initialized to a state that makes it ineffective. Further rendering will tend to bring the hierarchical depth buffer back to a more</p>	Format:	Enable
Format:	Enable		



3DSTATE_WM_HZ_OP													
		effective state.											
26	Pixel Position Offset Enable	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>Enable Enumerated Type</td> </tr> </table> <p>Enables the device to offset pixel positions by 0.5 both in horizontal and vertical directions.</p> <p style="text-align: center;">Programming Notes</p> <p>Setting this field along with setting the Pixel Location to upper left and number of multisamples to greater than one will cause the device to offset pixel positions by 0.5 both in horizontal and vertical directions. It is to be noted this is done to adjust the pixel co-ordinate system to DX9 like, so any WM_HZ_OP screen space rectangles (eg: legacy HiZ Clear, Resolve etc) generated internally by driver in this mode needs to be aware of this offset adjustment and send the rectangles according to alignment restriction taking this offset adjustment into consideration. SW can choose to set this bit only for DX9 API. DX10/OGL API's should not have any effect by setting or not setting this bit.</p>	Format:	Enable Enumerated Type									
Format:	Enable Enumerated Type												
25	Full Surface Depth and Stencil Clear	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p style="text-align: center;">Programming Notes</p> <p>Setting this field to "1" along with "Depth buffer clear" will cause all the pixels/samples in an the HZ and Stencil CLs to be cleared. Software must set this only when the APP requires the entire Depth surface to be cleared. Setting this field to "1" for STC-buffer only clear without "depth buffer clear" will cause all the pixels/samples in the STC-CL to get the stc-ref value.</p>	Format:	Enable									
Format:	Enable												
24	Reserved	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												
23:16	Stencil Clear Value	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U8.0</td> </tr> </table> <p>This field specifies the stencil clear value.</p>	Format:	U8.0									
Format:	U8.0												
15:13	Number of Multisamples	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U3 Enumerated Type</td> </tr> </table> <p>This field specifies how many samples/pixel exist in the Depth Buffer and Stencil buffers, as $\log_2(\#samples)$.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>1</td> <td>1 sample/pixel</td> </tr> <tr> <td>1h</td> <td>2</td> <td>2 samples/pixel</td> </tr> </tbody> </table>	Format:	U3 Enumerated Type	Value	Name	Description	0h	1	1 sample/pixel	1h	2	2 samples/pixel
Format:	U3 Enumerated Type												
Value	Name	Description											
0h	1	1 sample/pixel											
1h	2	2 samples/pixel											



3DSTATE_WM_HZ_OP				
		2h	4 samples/pixel	
		3h	8 samples/pixel	
		4h	16 samples/pixel	
		5h-7h	Reserved	
	12:0	Reserved		
<p style="text-align: center;">2</p> <p>Programming Notes: The clear rectangle x and y min and max values must be shifted by the LOD level; i.e. the hardware does not include the LOD in this function. Hence to clear any particular X, Y from the base level, to clear the contents at level "LOD" use (X»LOD) and (Y»LOD).</p>	31:16	Clear Rectangle Y Min		
		Format:	U16 in Pixels from Depth Buffer origin (upper left corner)	
			Specifies Ymin value of (inclusive) of clear rectangle with the Depth Buffer, used for clipping. Pixels with Y coordinates less than Ymin will not be affected.	
			Value	Name
		[0,16383]	(Device ignores bits 31:30)	
	15:0	Clear Rectangle X Min		
		Format:	U16 in Pixels from Depth Buffer origin (upper left corner)	
		Specifies Xmin value of (inclusive) of clear rectangle with the Depth Buffer, used for clipping. Pixels with X coordinates less than or equal to Xmin will not be affected.		
		Value	Name	
		[0,16383]	(Device ignores bits 15:14)	
<p style="text-align: center;">3</p> <p>Programming Notes: See the programming note in the previous DWORD for the Min values. The clear rectangle x and y min and max values must be shifted by the LOD level; i.e. the hardware does not include the LOD in this function. Hence to clear any particular X, Y from the base level, to clear the contents at level "LOD" use (X»LOD) and (Y»LOD). Hence the max values must be less than or equal to: (Surface Width » LOD) and (Surface Height » LOD) for X Max and Y Max respectively.</p>	31:16	Clear Rectangle Y Max		
		Format:	U16 in Pixels from Depth Buffer origin (lower right corner)	
			Specifies Ymax value of (exclusive) of clear rectangle with the Depth Buffer, used for clipping. Pixels with Y coordinates greater than Ymax will be not be cleared.	
			Value	Name
		[0,16383]	(Device ignores bits 31:30)	
	15:0	Clear Rectangle X Max		
		Format:	U16 in Pixels from Depth Buffer origin (lower right corner)	
		Specifies Xmax value of (exclusive) of clear rectangle with the Depth Buffer, used for clipping. Pixels with X coordinates greater than or equal to Xmax will be not be affected.		
		Value	Name	
		[0,16383]	(Device ignores bits 15:14)	



3DSTATE_WM_HZ_OP		
4	31:16	Reserved
		Format: MBZ
	15:0	Sample Mask
		Format: Right-justified bitmask (Bit 0 = Sample0). Number of bits that are used is determined by Num Multisamples (3DSTATE_WM_HZ_OP) A per-multisample-position mask state variable that is immediately and unconditionally ANDed with the sample coverage mask as part of the rasterization process. This mask is applied prior to centroid selection.
		Programming Notes
		If Number of Multisamples is NUMSAMPLES_1, bits 15:1 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_2, bits 15:2 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_4, bits 15:4 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_8, bits 15:8 of this field will be zeroed by HW.



A64 Byte Scaled Read MSD

MSD2R_A64_BS - A64 Byte Scaled Read MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Scaled R/W	
Group:	Byte Scaled R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_A64_MHP If present, modifies the address calculations.
	18:15	Message Type Default Value: 02h Format: Opcode A64 Scattered Read message
	14	Reserved Format: MBZ Ignored
	13	Invalidate After Read Format: MDC_IAR Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs
	12	SIMD Mode Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)
	11:10	Data Elements Format: MDC_A64_DS Specifies the number of data elements to be read or written
	9:8	A64 Scattered Message Subtype Default Value: 0h Format: Opcode Byte Read/Write subtype



MSD2R_A64_BS - A64 Byte Scaled Read MSD	
7:0	Sideband Scaled Offset Format: MDC_A64_SBSO In combination with Header Present field, specifies the Scale pitch and the Offset for the message.



A64 Byte Scaled Write MSD

MSD2W_A64_BS - A64 Byte Scaled Write MSD						
Source:	DataPort 2					
Length Bias:	1					
Family:	Scaled R/W					
Group:	Byte Scaled R/W					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_A64_MHP</td> </tr> </table> <p>If present, modifies the address calculations.</p>	Format:	MDC_A64_MHP		
	Format:	MDC_A64_MHP				
	18:15	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>0Bh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Scattered Write message</p>	Default Value:	0Bh	Format:	Opcode
	Default Value:	0Bh				
	Format:	Opcode				
	14	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
	Format:	MBZ				
	13	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
Format:	MBZ					
12	<p>SIMD Mode</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_SM2</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2			
Format:	MDC_SM2					
11:10	<p>Data Elements</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_A64_DS</td> </tr> </table> <p>Specifies the number of data elements to be read or written</p>	Format:	MDC_A64_DS			
Format:	MDC_A64_DS					
9:8	<p>A64 Scattered Message Subtype</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Byte Read/Write subtype</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					



MSD2W_A64_BS - A64 Byte Scaled Write MSD	
7:0	Sideband Scaled Offset Format: MDC_A64_SBSO In combination with Header Present field, specifies the Scale pitch and the Offset for the message.



A64 Byte Scattered Read MSD

MSD1R_A64_BS - A64 Byte Scattered Read MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Scattered R/W	
Group:	Byte Scattered R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type Default Value: 10h Format: Opcode A64 Scattered Read message
	13	Invalidate After Read Format: MDC_IAR Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs
	12	SIMD Mode Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)
	11:10	Data Elements Format: MDC_A64_DS Specifies the number of data elements to be read or written
	9:8	A64 Scattered Message Subtype Default Value: 0h Format: Opcode Byte Read/Write subtype
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Byte Scattered Write MSD

MSD1W_A64_BS - A64 Byte Scattered Write MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Scattered R/W	
Group:	Byte Scattered R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type Default Value: 1Ah Format: Opcode A64 Scattered Write message
	13	Reserved Format: MBZ Ignored
	12	SIMD Mode Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)
	11:10	Data Elements Format: MDC_A64_DS Specifies the number of data elements to be read or written
	9:8	A64 Scattered Message Subtype Default Value: 0h Format: Opcode Byte Read/Write subtype
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword Scaled Read MSD

MSD2R_A64_DWS - A64 Dword Scaled Read MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Scaled R/W	
Group:	DW Scaled R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_A64_MHP If present, modifies the address calculations.
	18:15	Message Type Default Value: 02h Format: Opcode A64 Scattered Read message
	14	Reserved Format: MBZ Ignored
	13	Invalidate After Read Format: MDC_IAR Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs
	12	SIMD Mode Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)
	11:10	Data Elements Format: MDC_A64_DS Specifies the number of data elements to be read or written
	9:8	A64 Scattered Message Subtype Default Value: 1h Format: Opcode Dword Read/Write subtype



MSD2R_A64_DWS - A64 Dword Scaled Read MSD

	7:0	Sideband Scaled Offset
		Format: MDC_A64_SBSO
		In combination with Header Present field, specifies the Scale pitch and the Offset for the message.



A64 Dword Scaled Write MSD

MSD2W_A64_DWS - A64 Dword Scaled Write MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Scaled R/W	
Group:	DW Scaled R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_A64_MHP If present, modifies the address calculations.
	18:15	Message Type Default Value: 0Bh Format: Opcode A64 Scattered Write message
	14	Reserved Format: MBZ Ignored
	13	Reserved Format: MBZ Ignored
	12	SIMD Mode Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)
	11:10	Data Elements Format: MDC_A64_DS Specifies the number of data elements to be read or written
	9:8	A64 Scattered Message Subtype Default Value: 1h Format: Opcode Dword Read/Write subtype



MSD2W_A64_DWS - A64 Dword Scaled Write MSD	
7:0	Sideband Scaled Offset
	Format: MDC_A64_SBSO
	In combination with Header Present field, specifies the Scale pitch and the Offset for the message.



A64 Dword Scattered Read MSD

MSD1R_A64_DWS - A64 Dword Scattered Read MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Scattered R/W					
Group:	DW Scattered R/W					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>Indicates that the message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>10h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Scattered Read message</p>	Default Value:	10h	Format:	Opcode
	Default Value:	10h				
	Format:	Opcode				
	13	<p>Invalidate After Read</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_IAR</td> </tr> </table> <p>Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs</p>	Format:	MDC_IAR		
	Format:	MDC_IAR				
	12	<p>SIMD Mode</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_SM2</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2		
Format:	MDC_SM2					
11:10	<p>Data Elements</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_A64_DS</td> </tr> </table> <p>Specifies the number of data elements to be read or written</p>	Format:	MDC_A64_DS			
Format:	MDC_A64_DS					
9:8	<p>A64 Scattered Message Subtype</p> <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Dword Read/Write subtype</p>	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
7:0	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword Scattered Write MSD

MSD1W_A64_DWS - A64 Dword Scattered Write MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Scattered R/W					
Group:	DW Scattered R/W					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_MHF</td> </tr> </table> Indicates that the message forbids a header	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td style="text-align: center;">1Ah</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> A64 Scattered Write message	Default Value:	1Ah	Format:	Opcode
	Default Value:	1Ah				
	Format:	Opcode				
	13	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table> Ignored	Format:	MBZ		
	Format:	MBZ				
	12	SIMD Mode <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_SM2</td> </tr> </table> Specifies the SIMD mode of the message (number of slots processed)	Format:	MDC_SM2		
Format:	MDC_SM2					
11:10	Data Elements <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_A64_DS</td> </tr> </table> Specifies the number of data elements to be read or written	Format:	MDC_A64_DS			
Format:	MDC_A64_DS					
9:8	A64 Scattered Message Subtype <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td style="text-align: center;">1h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Dword Read/Write subtype	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
7:0	Binding Table Index <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_STATELESS</td> </tr> </table> Specifies the message is stateless	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword SIMD4x2 Untyped Atomic Float Binary with Return Data Operation MSD

MSD1R_A64_DWAF2_4x2 - A64 Dword SIMD4x2 Untyped Atomic Float Binary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Float Binary Operation	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type
		Default Value: 1Eh
		Format: Opcode A64 Untyped Atomic Float Operation SIMD4x2 message
	13	Return Data Control
Default Value: 1h		
Format: Opcode Specifies that return data is sent back to the thread.		
12	Reserved	
	Format: MBZ Ignored	
11	Data Width	
	Default Value: 0h	
	Format: Opcode Operations are on 32-bit floats.	
10	Reserved	
	Format: MBZ Ignored	



MSD1R_A64_DWAF2_4x2 - A64 Dword SIMD4x2 Untyped Atomic Float Binary with Return Data Operation MSD		
	9:8	Atomic Float Operation Format: MDC_FOP2 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword SIMD4x2 Untyped Atomic Float Binary Write Only Operation MSD

MSD1W_A64_DWAF2_4x2 - A64 Dword SIMD4x2 Untyped Atomic Float Binary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Float Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type Default Value: 1Eh Format: Opcode A64 Untyped Atomic Float Operation SIMD4x2 message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11	Data Width Default Value: 0h Format: Opcode Operations are on 32-bit floats.
	10	Reserved Format: MBZ Ignored



MSD1W_A64_DWAF2_4x2 - A64 Dword SIMD4x2 Untyped Atomic Float Binary Write Only Operation MSD		
	9:8	Atomic Float Operation Type Format: MDC_FOP2 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword SIMD4x2 Untyped Atomic Float Trinary with Return Data Operation MSD

MSD1R_A64_DWAF3_4x2 - A64 Dword SIMD4x2 Untyped Atomic Float Trinary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Float Trinary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> Indicates that the message forbids a header	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>1Eh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> A64 Untyped Atomic Float Operation SIMD4x2 message	Default Value:	1Eh	Format:	Opcode
	Default Value:	1Eh				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that return data is sent back to the thread.	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
11	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Operations are on 32-bit floats.	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
10	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					



MSD1R_A64_DWAF3_4x2 - A64 Dword SIMD4x2 Untyped Atomic Float Trinary with Return Data Operation MSD		
	9:8	Atomic Float Operation Format: MDC_FOP3 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword SIMD4x2 Untyped Atomic Float Trinary Write Only Operation MSD

MSD1W_A64_DWAF3_4x2 - A64 Dword SIMD4x2 Untyped Atomic Float Trinary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Float Trinary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> Indicates that the message forbids a header	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>1Eh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> A64 Untyped Atomic Float Operation SIMD4x2 message	Default Value:	1Eh	Format:	Opcode
	Default Value:	1Eh				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
11	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Operations are on 32-bit floats.	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
10	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					



MSD1W_A64_DWAF3_4x2 - A64 Dword SIMD4x2 Untyped Atomic Float Trinary Write Only Operation MSD		
	9:8	Atomic Float Operation Format: MDC_FOP3 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword SIMD4x2 Untyped Atomic Integer Binary with Return Data Operation MSD

MSD1R_A64_DWAI2_4x2 - A64 Dword SIMD4x2 Untyped Atomic Integer Binary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation SIMD4x2 message</p>	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that return data is sent back to the thread.</p>	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 32-bit integers</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword SIMD4x2 Untyped Atomic Integer Binary Write Only Operation MSD

MSD1W_A64_DWAI2_4x2 - A64 Dword SIMD4x2 Untyped Atomic Integer Binary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation SIMD4x2 message</p>	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 32-bit integers</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword SIMD4x2 Untyped Atomic Integer Trinary with Return Data Operation MSD

MSD1R_A64_DWAI3_4x2 - A64 Dword SIMD4x2 Untyped Atomic Integer Trinary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Trinary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> The message forbids a header	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> A64 Untyped Atomic Integer Operation SIMD4x2 message	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that return data is sent back to the thread.	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Operations are on 32-bit integers	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> Specifies the message is stateless	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword SIMD4x2 Untyped Atomic Integer Trinary Write Only Operation MSD

MSD1W_A64_DWAI3_4x2 - A64 Dword SIMD4x2 Untyped Atomic Integer Trinary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Trinary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation SIMD4x2 message</p>	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 32-bit integers</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword SIMD4x2 Untyped Atomic Integer Unary with Return Data Operation MSD

MSD1R_A64_DWAI1_4x2 - A64 Dword SIMD4x2 Untyped Atomic Integer Unary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Unary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF The message forbids a header
	18:14	Message Type Default Value: 13h Format: Opcode A64 Untyped Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Data Width Default Value: 0h Format: Opcode Operations are on 32-bit integers
	11:8	Atomic Integer Operation Format: MDC_AOP1 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword SIMD4x2 Untyped Atomic Integer Unary Write Only Operation MSD

MSD1W_A64_DWAI1_4x2 - A64 Dword SIMD4x2 Untyped Atomic Integer Unary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation SIMD4x2 message</p>	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 32-bit integers</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP1</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword Untyped Atomic Float Binary with Return Data Operation MSD

MSD1R_A64_DWAF2 - A64 Dword Untyped Atomic Float Binary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Float Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type Default Value: 1Dh Format: Opcode A64 Untyped Atomic Float Operation message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	SIMD Mode Format: MDC_SM2S Only SIMD8 operations are supported.
	11	Data Width Default Value: 0h Format: Opcode Operations are on 32-bit floats.
	10	Reserved Format: MBZ Ignored



MSD1R_A64_DWAF2 - A64 Dword Untyped Atomic Float Binary with Return Data Operation MSD

	9:8	Atomic Float Operation Format: MDC_FOP2 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword Untyped Atomic Float Binary Write Only Operation MSD

MSD1W_A64_DWAF2 - A64 Dword Untyped Atomic Float Binary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Float Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> Indicates that the message forbids a header	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>1Dh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> A64 Untyped Atomic Float Operation message	Default Value:	1Dh	Format:	Opcode
	Default Value:	1Dh				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	SIMD Mode <table border="1"> <tr> <td>Format:</td> <td>MDC_SM2S</td> </tr> </table> Only SIMD8 operations are supported.	Format:	MDC_SM2S			
Format:	MDC_SM2S					
11	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Operations are on 32-bit floats.	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
10	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					



MSD1W_A64_DWAF2 - A64 Dword Untyped Atomic Float Binary Write Only Operation MSD		
	9:8	Atomic Float Operation Type Format: MDC_FOP2 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword Untyped Atomic Float Trinary with Return Data Operation MSD

MSD1R_A64_DWAF3 - A64 Dword Untyped Atomic Float Trinary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Float Trinary Operation	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type
		Default Value: 1Dh
		Format: Opcode A64 Untyped Atomic Float Operation message
	13	Return Data Control
Default Value: 1h		
Format: Opcode Specifies that return data is sent back to the thread.		
12	SIMD Mode	
	Format: MDC_SM2S Only SIMD8 operations are supported.	
11	Data Width	
	Default Value: 0h	
	Format: Opcode Operations are on 32-bit floats.	
10	Reserved	
	Format: MBZ Ignored	



MSD1R_A64_DWAF3 - A64 Dword Untyped Atomic Float Trinary with Return Data Operation MSD		
	9:8	Atomic Float Operation Format: MDC_FOP3 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword Untyped Atomic Float Trinary Write Only Operation MSD

MSD1W_A64_DWAF3 - A64 Dword Untyped Atomic Float Trinary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Float Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type Default Value: 1Dh Format: Opcode A64 Untyped Atomic Float Operation message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	SIMD Mode Format: MDC_SM2S Only SIMD8 operations are supported.
	11	Data Width Default Value: 0h Format: Opcode Operations are on 32-bit floats.
	10	Reserved Format: MBZ Ignored



MSD1W_A64_DWAF3 - A64 Dword Untyped Atomic Float Trinary Write Only Operation MSD		
	9:8	Atomic Float Operation Format: MDC_FOP3 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword Untyped Atomic Integer Binary with Return Data Operation MSD

MSD1R_A64_DWAI2 - A64 Dword Untyped Atomic Integer Binary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> The message forbids a header	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>12h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> A64 Untyped Atomic Integer Operation message	Default Value:	12h	Format:	Opcode
	Default Value:	12h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that return data is sent back to the thread.	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Operations are on 32-bit integers	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> Specifies the message is stateless	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword Untyped Atomic Integer Binary Write Only Operation MSD

MSD1W_A64_DWAI2 - A64 Dword Untyped Atomic Integer Binary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>12h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation message</p>	Default Value:	12h	Format:	Opcode
	Default Value:	12h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 32-bit integers</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword Untyped Atomic Integer Trinary with Return Data Operation MSD

MSD1R_A64_DWAI3 - A64 Dword Untyped Atomic Integer Trinary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF The message forbids a header
	18:14	Message Type Default Value: 12h Format: Opcode A64 Untyped Atomic Integer Operation message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Data Width Default Value: 0h Format: Opcode Operations are on 32-bit integers
	11:8	Atomic Integer Operation Format: MDC_AOP3 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword Untyped Atomic Integer Trinary Write Only Operation MSD

MSD1W_A64_DWAI3 - A64 Dword Untyped Atomic Integer Trinary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF The message forbids a header
	18:14	Message Type Default Value: 12h Format: Opcode A64 Untyped Atomic Integer Operation message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Data Width Default Value: 0h Format: Opcode Operations are on 32-bit integers
	11:8	Atomic Integer Operation Format: MDC_AOP3 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Dword Untyped Atomic Integer Unary with Return Data Operation MSD

MSD1R_A64_DWAI1 - A64 Dword Untyped Atomic Integer Unary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> The message forbids a header	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>12h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> A64 Untyped Atomic Integer Operation message	Default Value:	12h	Format:	Opcode
	Default Value:	12h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that return data is sent back to the thread.	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Operations are on 32-bit integers	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP1</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> Specifies the message is stateless	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Dword Untyped Atomic Integer Unary Write Only Operation MSD

MSD1W_A64_DWAI1 - A64 Dword Untyped Atomic Integer Unary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>12h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation message</p>	Default Value:	12h	Format:	Opcode
	Default Value:	12h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 32-bit integers</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP1</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Hword Block Read MSD

MSD1R_A64_HWB - A64 Hword Block Read MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Block R/W					
Group:	HW Block R/W					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHR</td> </tr> </table> <p>Indicates that the message requires a header.</p>	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18:14	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>14h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Oword Block Read message</p>	Default Value:	14h	Format:	Opcode
	Default Value:	14h				
	Format:	Opcode				
	13	<p>Invalidate After Read</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_IAR</td> </tr> </table> <p>Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs</p>	Format:	MDC_IAR		
Format:	MDC_IAR					
12:11	<p>A64 Block Message Subtype</p> <table border="1"> <tr> <td>Default Value:</td> <td>3h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Hword Block Read/Write subtype</p>	Default Value:	3h	Format:	Opcode	
Default Value:	3h					
Format:	Opcode					
10:8	<p>Data Elements</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_A64_DB_HW</td> </tr> </table> <p>Specifies the number of contiguous Hwords to be read or written</p>	Format:	MDC_A64_DB_HW			
Format:	MDC_A64_DB_HW					
7:0	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Hword Block Write MSD

MSD1W_A64_HWB - A64 Hword Block Write MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Block R/W	
Group:	HW Block R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHR Indicates that the message requires a header.
	18:14	Message Type
		Default Value: 15h Format: Opcode A64 Hword Block Write message
	13	Reserved
		Format: MBZ Ignored
12:11	A64 Block Message Subtype	
	Default Value: 3h Format: Opcode Hword Block Read/Write subtype	
10:8	Data Elements	
	Format: MDC_A64_DB_HW Specifies the number of contiguous Hwords to be read or written	
7:0	Binding Table Index	
	Format: MDC_STATELESS Specifies the message is stateless	



A64 Oword Block Read MSD

MSD1R_A64_OWB - A64 Oword Block Read MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Block R/W	
Group:	OW Block R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHR Indicates that the message requires a header.
	18:14	Message Type
		Default Value: 14h
		Format: Opcode A64 Oword Block Read message
	13	Invalidate After Read
Format: MDC_IAR Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs		
12:11	A64 Block Message Subtype	
	Default Value: 0h	
	Format: Opcode Oword Block Read/Write subtype	
10:8	Data Elements	
	Format: MDC_A64_DB_OW Specifies the number of contiguous Owords to be read or written	
7:0	Binding Table Index	
	Format: MDC_STATELESS Specifies the message is stateless	



A64 Oword Block Write MSD

MSD1W_A64_OWB - A64 Oword Block Write MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Block R/W	
Group:	OW Block R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header.
	18:14	Message Type Default Value: 15h Format: Opcode A64 Oword Block Write message
	13	Reserved Format: MBZ Ignored
	12:11	A64 Block Message Subtype Default Value: 0h Format: Opcode Oword Block Read/Write subtype
	10:8	Data Elements Format: MDC_A64_DB_OW Specifies the number of contiguous Owords to be read or written
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Oword Dual Block Read MSD

MSD1R_A64_OWDB - A64 Oword Dual Block Read MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Block R/W	
Group:	OW Dual Block R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHR Indicates that the message requires a header.
	18:14	Message Type
		Default Value: 14h
		Format: Opcode A64 Oword Block Read message
	13	Invalidate After Read
Format: MDC_IAR Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs		
12:11	A64 Block Message Subtype	
	Default Value: 2h	
	Format: Opcode Oword Dual Block Read/Write subtype	
10:8	Data Elements	
	Format: MDC_A64_DB_OWD Specifies the number of contiguous Owords to be read or written	
7:0	Binding Table Index	
	Format: MDC_STATELESS Specifies the message is stateless	



A64 Oword Dual Block Write MSD

MSD1W_A64_OWDB - A64 Oword Dual Block Write MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Block R/W	
Group:	OW Dual Block R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHR Indicates that the message requires a header.
	18:14	Message Type
		Default Value: 15h Format: Opcode A64 Oword Block Write message
	13	Reserved
		Format: MBZ Ignored
12:11	A64 Block Message Subtype	
	Default Value: 2h Format: Opcode Oword Dual Block Read/Write subtype	
10:8	Data Elements	
	Format: MDC_A64_DB_OWD Specifies the number of contiguous Owords to be read or written	
7:0	Binding Table Index	
	Format: MDC_STATELESS Specifies the message is stateless	



A64 Oword Unaligned Block Read MSD

MSD1R_A64_OWUB - A64 Oword Unaligned Block Read MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Block R/W	
Group:	OW Unaligned Block R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHR Indicates that the message requires a header.
	18:14	Message Type
		Default Value: 14h
		Format: Opcode A64 Oword Block Read message
	13	Reserved
Format: MBZ Ignored		
12:11	A64 Block Message Subtype	
	Default Value: 1h	
	Format: Opcode Oword Unaligned Block Read subtype	
10:8	Data Elements	
	Format: MDC_A64_DB_OW Specifies the number of contiguous Owords to be read	
7:0	Binding Table Index	
	Format: MDC_STATELESS Specifies the message is stateless	



A64 Qword Scaled Read MSD

MSD2R_A64_QWS - A64 Qword Scaled Read MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Scaled R/W	
Group:	QW Scaled R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_A64_MHP If present, modifies the address calculations.
	18:15	Message Type Default Value: 02h Format: Opcode A64 Scattered Read message
	14	Reserved Format: MBZ Ignored
	13	Invalidate After Read Format: MDC_IAR Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs
	12	SIMD Mode Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)
	11:10	Data Elements Format: MDC_A64_DS Specifies the number of data elements to be read or written
	9:8	A64 Scattered Message Subtype Default Value: 2h Format: Opcode Qword Read/Write subtype



MSD2R_A64_QWS - A64 Qword Scaled Read MSD			
7:0	<p>Sideband Scaled Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_A64_SBSO</td> </tr> </table> <p>In combination with Header Present field, specifies the Scale pitch and the Offset for the message.</p>	Format:	MDC_A64_SBSO
Format:	MDC_A64_SBSO		



A64 Qword Scaled Write MSD

MSD2W_A64_QWS - A64 Qword Scaled Write MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Scaled R/W	
Group:	QW Scaled R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_A64_MHP If present, modifies the address calculations.
	18:15	Message Type Default Value: 0Bh Format: Opcode A64 Scattered Write message
	14	Reserved Format: MBZ Ignored
	13	Reserved Format: MBZ Ignored
	12	SIMD Mode Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)
	11:10	Data Elements Format: MDC_A64_DS Specifies the number of data elements to be read or written
	9:8	A64 Scattered Message Subtype Default Value: 2h Format: Opcode Qword Read/Write subtype



MSD2W_A64_QWS - A64 Qword Scaled Write MSD			
7:0	<p>Sideband Scaled Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_A64_SBSO</td> </tr> </table> <p>In combination with Header Present field, specifies the Scale pitch and the Offset for the message.</p>	Format:	MDC_A64_SBSO
Format:	MDC_A64_SBSO		



A64 Qword Scattered Read MSD

MSD1R_A64_QWS - A64 Qword Scattered Read MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Scattered R/W	
Group:	QW Scattered R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type
		Default Value: 10h
		Format: Opcode A64 Scattered Read message
	13	Invalidate After Read
		Format: MDC_IAR Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs
	12	SIMD Mode
Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)		
11:10	Data Elements	
	Format: MDC_A64_DS Specifies the number of data elements to be read or written	
9:8	A64 Scattered Message Subtype	
	Default Value: 2h	
	Format: Opcode Qword Read/Write subtype	
7:0	Binding Table Index	
	Format: MDC_STATELESS Specifies the message is stateless	



A64 Qword Scattered Write MSD

MSD1W_A64_QWS - A64 Qword Scattered Write MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Scattered R/W	
Group:	QW Scattered R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type Default Value: 1Ah Format: Opcode A64 Scattered Write message
	13	Reserved Format: MBZ Ignored
	12	SIMD Mode Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)
	11:10	Data Elements Format: MDC_A64_DS Specifies the number of data elements to be read or written
	9:8	A64 Scattered Message Subtype Default Value: 2h Format: Opcode Qword Read/Write subtype
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Qword SIMD4x2 Untyped Atomic Integer Binary with Return Data Operation MSD

MSD1R_A64_QWAI2_4x2 - A64 Qword SIMD4x2 Untyped Atomic Integer Binary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Qword Untyped Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation SIMD4x2 message</p>	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that return data is sent back to the thread.</p>	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 64-bit integers</p>	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Qword SIMD4x2 Untyped Atomic Integer Binary Write Only Operation MSD

MSD1W_A64_QWAI2_4x2 - A64 Qword SIMD4x2 Untyped Atomic Integer Binary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Qword Untyped Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation SIMD4x2 message</p>	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 64-bit integers</p>	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Qword SIMD4x2 Untyped Atomic Integer Trinary with Return Data Operation MSD

MSD1R_A64_QWAI3_4x2 - A64 Qword SIMD4x2 Untyped Atomic Integer Trinary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Qword Untyped Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHF The message forbids a header
	18:14	Message Type
		Default Value: 13h
		Format: Opcode A64 Untyped Atomic Integer Operation SIMD4x2 message
	13	Return Data Control
		Default Value: 1h
Format: Opcode Specifies that return data is sent back to the thread.		
12	Data Width	
	Default Value: 1h	
	Format: Opcode Operations are on 64-bit integers	
11:8	Atomic Integer Operation	
	Specifies the atomic integer operation to be performed.	
	Workaround Workaround : CMPWR_2W is not supported in A64 SIMD4x2	
7:0	Binding Table Index	
	Format: MDC_STATELESS Specifies the message is stateless	



A64 Qword SIMD4x2 Untyped Atomic Integer Trinary Write Only Operation MSD

MSD1W_A64_QWAI3_4x2 - A64 Qword SIMD4x2 Untyped Atomic Integer Trinary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Qword Untyped Atomic Integer Trinary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation SIMD4x2 message</p>	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 64-bit integers</p>	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Qword SIMD4x2 Untyped Atomic Integer Unary with Return Data Operation MSD

MSD1R_A64_QWAI1_4x2 - A64 Qword SIMD4x2 Untyped Atomic Integer Unary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Qword Untyped Atomic Integer Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation SIMD4x2 message</p>	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that return data is sent back to the thread.</p>	Default Value:	1h	Format:	Opcode
	Default Value:	1h				
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 64-bit integers</p>	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP1</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Qword SIMD4x2 Untyped Atomic Integer Unary Write Only Operation MSD

MSD1W_A64_QWAI1_4x2 - A64 Qword SIMD4x2 Untyped Atomic Integer Unary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Qword Untyped Atomic Integer Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>13h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation SIMD4x2 message</p>	Default Value:	13h	Format:	Opcode
	Default Value:	13h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 64-bit integers</p>	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP1</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Qword Untyped Atomic Integer Binary with Return Data Operation MSD

MSD1R_A64_QWAI2 - A64 Qword Untyped Atomic Integer Binary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Qword Untyped Atomic Integer Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF The message forbids a header
	18:14	Message Type Default Value: 12h Format: Opcode A64 Untyped Atomic Integer Operation message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Data Width Default Value: 1h Format: Opcode Operations are on 64-bit integers
	11:8	Atomic Integer Operation Format: MDC_AOP2 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Qword Untyped Atomic Integer Binary Write Only Operation MSD

MSD1W_A64_QWAI2 - A64 Qword Untyped Atomic Integer Binary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Qword Untyped Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> The message forbids a header	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>12h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> A64 Untyped Atomic Integer Operation message	Default Value:	12h	Format:	Opcode
	Default Value:	12h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Operations are on 64-bit integers	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> Specifies the message is stateless	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Qword Untyped Atomic Integer Trinary with Return Data Operation MSD

MSD1R_A64_QWAI3 - A64 Qword Untyped Atomic Integer Trinary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Qword Untyped Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF The message forbids a header
	18:14	Message Type Default Value: 12h Format: Opcode A64 Untyped Atomic Integer Operation message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Data Width Default Value: 1h Format: Opcode Operations are on 64-bit integers
	11:8	Atomic Integer Operation Format: MDC_AOP3S Specifies the atomic integer operation to be performed. Workaround Workaround : CMPWR_2W is not supported in A64 QWord SIMD8.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Qword Untyped Atomic Integer Trinary Write Only Operation MSD

MSD1W_A64_QWAI3 - A64 Qword Untyped Atomic Integer Trinary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Qword Untyped Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF The message forbids a header
	18:14	Message Type Default Value: 12h Format: Opcode A64 Untyped Atomic Integer Operation message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Data Width Default Value: 1h Format: Opcode Operations are on 64-bit integers
	11:8	Atomic Integer Operation Format: MDC_AOP3 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Qword Untyped Atomic Integer Unary with Return Data Operation MSD

MSD1R_A64_QWAI1 - A64 Qword Untyped Atomic Integer Unary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Qword Untyped Atomic Integer Unary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF The message forbids a header
	18:14	Message Type Default Value: 12h Format: Opcode A64 Untyped Atomic Integer Operation message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Data Width Default Value: 1h Format: Opcode Operations are on 64-bit integers
	11:8	Atomic Integer Operation Format: MDC_AOP1 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Qword Untyped Atomic Integer Unary Write Only Operation MSD

MSD1W_A64_QWAI1 - A64 Qword Untyped Atomic Integer Unary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Qword Untyped Atomic Integer Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHF</td> </tr> </table> <p>The message forbids a header</p>	Format:	MDC_MHF		
	Format:	MDC_MHF				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>12h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>A64 Untyped Atomic Integer Operation message</p>	Default Value:	12h	Format:	Opcode
	Default Value:	12h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 64-bit integers</p>	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP1</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_STATELESS</td> </tr> </table> <p>Specifies the message is stateless</p>	Format:	MDC_STATELESS			
Format:	MDC_STATELESS					



A64 Scaled Untyped Surface Read MSD

MSD2R_A64_US - A64 Scaled Untyped Surface Read MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Untyped Surface R/W	
Group:	Scaled Untyped Surface R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_A64_MHP If present, modifies the address calculations.
		Message Type Default Value: 03h Format: Opcode A64 Untyped Surface Read message
	14	Reserved Format: MBZ Ignored
		SIMD Mode Format: MDC_SM3 Specifies the SIMD mode of the message (number of slots processed)
	11:8	Channel Mask Format: MDC_CMASK Specifies which RGBA channels are included in the message payload.
		Sideband Scaled Offset Format: MDC_A64_SBSO In combination with Header Present field, specifies the Scale pitch and the Offset for the message.



A64 Scaled Untyped Surface Write MSD

MSD2W_A64_US - A64 Scaled Untyped Surface Write MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Untyped Surface R/W	
Group:	Scaled Untyped Surface R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_A64_MHP If present, modifies the address calculations.
	18:15	Message Type Default Value: 0Ah Format: Opcode A64 Untyped Surface Write message
	14	Reserved Format: MBZ Ignored
	13:12	SIMD Mode Format: MDC_SM3 Specifies the SIMD mode of the message (number of slots processed)
	11:8	Channel Mask Format: MDC_UW_CMASK Specifies which RGBA channels are included in the message payload.
	7:0	Sideband Scaled Offset Format: MDC_A64_SBSO In combination with Header Present field, specifies the Scale pitch and the Offset for the message.



A64 Untyped Surface Read MSD

MSD1R_A64_US - A64 Untyped Surface Read MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Surface R/W	
Group:	Scattered Untyped Surface R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF Indicates that the message forbids a header
		Message Type Default Value: 11h Format: Opcode A64 Untyped Surface Read message
	13:12	SIMD Mode Format: MDC_SM3 Specifies the SIMD mode of the message (number of slots processed)
		Channel Mask Format: MDC_CMASK Specifies which RGBA channels are included in the message payload.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



A64 Untyped Surface Write MSD

MSD1W_A64_US - A64 Untyped Surface Write MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Surface R/W	
Group:	Scattered Untyped Surface R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF Indicates that the message forbids a header
	18:14	Message Type Default Value: 19h Format: Opcode A64 Untyped Surface Write message
	13:12	SIMD Mode Format: MDC_SM3 Specifies the SIMD mode of the message (number of slots processed)
	11:8	Channel Mask Format: MDC_UW_CMASK Specifies which RGBA channels are included in the message payload.
	7:0	Binding Table Index Format: MDC_STATELESS Specifies the message is stateless



Addition

add - Addition			
Source:	Eulsa		
Length Bias:	4		
<p>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).</p>			
<p>Format: [(pred)] add[.cmod] (exec_size) dst src0 src1</p>			
Programming Notes			
Use a source modifier with add to implement subtraction.			
Syntax			
<pre>[(pred)] add[.cmod] (exec_size) reg reg reg [(pred)] add[.cmod] (exec_size) reg reg imm32</pre>			
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	Saturation	Source Modifier
Y	Y	Y	Y
Src Types		Dst Types	
*B,*W,*D		*B,*W,*D	
*B,*W,*D		F	
F		F	
DF		DF	
HF		HF	
*B,*W,*D		HF	
*W,*D		*W,*D	
*W,*D		*Q	
*Q		*W,*D	
*Q		*Q	
HF, F		HF, F	
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	!([RegSource][Src1.RegFile]='IMM')



add - Addition			
		Format:	EU_INSTRUCTION_SOURCES_REG_REG
	127:64	ImmSource	
		Exists If:	([ImmSource][Src1.RegFile]='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_IMM
	63:32	Operand Controls	
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header	
		Format:	EU_INSTRUCTION_HEADER



Addition with Carry

addc - Addition with Carry			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p>			
Format:	[(pred)] addc[.cmod] (exec_size) dst src0 src1		
Restriction			
Restriction : AccWrEn is required. The accumulator is an implicit destination and thus cannot be an explicit destination operand.			
Syntax			
[(pred)] addc[.cmod] (exec_size) reg reg reg [(pred)] addc[.cmod] (exec_size) reg reg imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src0.chan[n] + src1.chan[n]; acc.chan[n] = carry(src0.chan[n] + src1.chan[n]); } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	N	N
Src Types	Dst Types		
UD	UD		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource	
Exists If:		([ImmSource][Src1.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_REG_IMM		
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Arithmetic Shift Right

asr - Arithmetic Shift Right			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>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.</p> <p>In QWord mode, the shift count is taken from the low six bits of src1 regardless of the src1 type and treated as an unsigned integer in the range 0 to 63. Otherwise 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. The operation uses QWord mode if src0 or dst has the Q or UQ type but not if src1 is the only operand with the Q or UQ type.</p> <p>For positive values, this operation is $\text{src0} / 2^{\text{shiftCount}}$ and for negative values, this operation is $\text{src0} / 2^{\text{shiftCount} - 1}$.</p>			
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 imm32			
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 shiftCnt = src0 or dst has Q or UQ type ? src1.chan[n] & 0x3F : src1.chan[n] & 0x1F 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 Types		Dst Types	
*B,*W,*D		*B,*W,*D	
*W,*D		*W,*D	
*W,*D		*Q	



asr - Arithmetic Shift Right		
*Q		*W,*D
*Q		*Q
DWord	Bit	Description
0..3	127:64	RegSource
		Exists If: ([RegSource][Src1.RegFile]!='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile]='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
31:0	Header	
	Format: EU_INSTRUCTION_HEADER	



Average

avg - Average			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p>			
<p>Format: 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.</p>			
Syntax			
<pre>[(pred)] avg[.cmod] (exec_size) reg reg reg [(pred)] avg[.cmod] (exec_size) reg reg imm32</pre>			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = (src0.chan[n] + src1.chan[n] + 1) » 1; // Use arithmetic shift right. } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
*B,*W,*D	*B,*W,*D		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	((RegSource)[Src1.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG_REG	
127:64	ImmSource		
	Exists If:	((ImmSource)[Src1.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_REG_IMM		
63:32	Operand Controls		
Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
31:0	Header		
Format:	EU_INSTRUCTION_HEADER		



Bit Field Extract

bfe - Bit Field Extract			
Source:	Eulsa		
Length Bias:	4		
<p>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).</p> <p>The width and offset values are from the low five bits of src0 and src1 respectively, or src0 & 0x1f and src1 & 0x1f.</p> <p>If width is zero, the result is zero.</p> <p>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.</p>			
<p>Format:</p> <p>[(pred)] bfe (exec_size) dst src0 src1 src2</p>			
Restriction			
Restriction : No accumulator access, implicit or explicit.			
Restriction : All three-source instructions have certain restrictions, described in Instruction Formats.			
Syntax			
[(pred)] bfe (exec_size) reg reg reg reg			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { UD width = src0.chan[n][4:0]; UD offset = src1.chan[n][4:0]; if (width == 0) { dst.chan[n] = 0x00000000; } else if ((width + offset) < 32) { dst.chan[n] = src2.chan[n] << (32 - width - offset); if (src2 is signed) { dst.chan[n] = dst.chan[n] >> (32 - width); // pad sign bit of dst.chan } else { dst.chan[n] = dst.chan[n] >> (32 - width); // pad 0 } } else { if (src2 is signed) { dst.chan[n] = src2.chan[n] >> offset; // pad sign bit } else { dst.chan[n] = src2.chan[n] >> offset; // pad 0 } } } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types	Dst Types		
UD	UD		
D	D		
DWord	Bit	Description	
0..3	127:126	Reserved	



bfe - Bit Field Extract

	Format:	MBZ
125:106	Source 2	
	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC
105	Reserved	
	Format:	MBZ
104:85	Source 1	
	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC
84	Reserved	
	Format:	MBZ
83:64	Source 0	
	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC
63:56	Destination Register Number	
	Format:	DstRegNum
55:53	Destination Subregister Number	
	Format:	DstSubRegNum[2:0]
52:49	Destination Channel Enable	
	Format:	ChanEn[4]
	<p>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</p>	
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	
	Value	Name
	00b	Single Precision Float
	01b	DWord
	10b	Unsigned DWord
	11b	Double Precision Float



bfe - Bit Field Extract		
48:42	Reserved	
	Format:	MBZ
43:42	Source Data Type This field contains the data type for all three sources	
	Value	Name
	00b	Single Precision Float
	01b	DWord
	10b	Unsigned DWord
11b	Double Precision Float	
41:40	Source 2 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
39:38	Source 1 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
41:36	Reserved	
	Exists If:	/// ([Property[Source Modifier] == 'false')
	Format:	MBZ
37:36	Source 0 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
35	Reserved	
	Format:	MBZ
34	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	Destination Register File	
	Value	Name
	0	GRF
	1	MRF



bfe - Bit Field Extract		
	32	Reserved
		Format: MBZ
	31:0	Header
		Format: EU_INSTRUCTION_HEADER



Bit Field Insert 1

bfi1 - Bit Field Insert 1			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <pre>bfi dst src0 src1 src2 src3 // Translates to these two instructions: bfi1 dst src0 src1 bfi2 dst dst src2 src3</pre>			
Format: [(pred)] bfi1 (exec_size) dst src0 src1			
Programming Notes			
No accumulator access, implicit or explicit.			
Syntax			
[(pred)] bfi1 (exec_size) reg reg reg [(pred)] bfi1 (exec_size) reg reg imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { UD width = src0.chan[n][4:0]; UD offset = src1.chan[n][4:0]; dst = ((1 << width) - 1) << offset; } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types	Dst Types		
UD	UD		
D	D		
DWord	Bit	Description	



bfi1 - Bit Field Insert 1		
0..3	127:64	RegSource
		Exists If: ([RegSource][Src1.RegFile]!='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile]='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
31:0	Header	
	Format: EU_INSTRUCTION_HEADER	



Bit Field Insert 2

bfi2 - Bit Field Insert 2			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <pre>bfi dst src0 src1 src2 src3 // Translates to these two instructions: bfi1 dst src0 src1 bfi2 dst dst src2 src3</pre>			
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 Formats.			
Syntax			
[(pred)] bfi2 (exec_size) reg reg reg reg			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { UD offset = 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]); } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types	Dst Types		
UD	UD		
D	D		



DWord	Bit	Description								
0..3	127:126	Reserved Format: MBZ								
	125:106	Source 2 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC								
	105	Reserved Format: MBZ								
	104:85	Source 1 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC								
	84	Reserved Format: MBZ								
	83:64	Source 0 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC								
	63:56	Destination Register Number Format: DstRegNum								
	55:53	Destination Subregister 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 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 Type This field contains the data type for the destination <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Single Precision Float</td> </tr> <tr> <td>01b</td> <td>DWord</td> </tr> <tr> <td>10b</td> <td>Unsigned DWord</td> </tr> </tbody> </table>	Value	Name	00b	Single Precision Float	01b	DWord	10b	Unsigned DWord
	Value	Name								
00b	Single Precision Float									
01b	DWord									
10b	Unsigned DWord									



bfi2 - Bit Field Insert 2		
	11b	Double Precision Float
48:42	Reserved	
	Format:	MBZ
43:42	Source Data Type This field contains the data type for all three sources	
	Value	Name
	00b	Single Precision Float
	01b	DWord
	10b	Unsigned DWord
	11b	Double Precision Float
41:40	Source 2 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
39:38	Source 1 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
41:36	Reserved	
	Exists If:	/// ([Property[Source Modifier] == 'false')
	Format:	MBZ
37:36	Source 0 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
35	Reserved	
	Format:	MBZ
34	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	Destination Register File	
	Value	Name
	0	GRF
	1	MRF



bfi2 - Bit Field Insert 2		
	32	Reserved
		Format: MBZ
	31:0	Header
		Format: EU_INSTRUCTION_HEADER



Bit Field Reverse

bfrev - Bit Field Reverse			
Source:	Eulsa		
Length Bias:	4		
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			
Restriction : No accumulator access, implicit or explicit.			
Syntax			
[(pred)] bfrev (exec_size) reg reg [(pred)] bfrev (exec_size) reg imm32			
Pseudocode			
Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { for (idx = 0; idx < 32; idx++) { dst.chan[n][idx] = src0.chan[n][31-idx]; } } }			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types	Dst Types		
UD	UD		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	[[Operand Controls][Src0.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource	
Exists If:		[[Operand Controls][Src0.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Branch Converging

brc - Branch Converging					
Source:	Eulsa				
Length Bias:	4				
Description					
<p>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.</p> <p>UIP should reference the instruction where all channels are expected to come together. JIP should reference the end of the innermost conditional block.</p> <p>In GEN binary, JIP and UIP use locations src1 and src0 respectively when immediate and location src0 when reg64, where reg64 is accessed as paired DWord (regioning being <2;2,1>). dst must be IP. When the offsets are immediate, src0 regfile must be immediate.</p>					
Format:	[[pred]] brc (exec_size) JIP UIP				
Restriction					
Restriction : A brc instruction cannot use the Switch instruction option.					
Syntax					
[[pred]] brc (exec_size) imm32 imm32 [[pred]] brc (exec_size) reg64					
Pseudocode					
<pre>Evaluate(WrEn); for (n = 0; n < 32; n++) { if (WrEn[n]) { PcIP[n] = IP + UIP; } else { PcIP[n] = IP + 1; } } if (all PcIP != IP + 1) { // for all channels Jump(IP + JIP); }</pre>					
Predication	Conditional Modifier	Saturation	Source Modifier		
Y	N	N	N		
Src Types					
D					
DWord	Bit	Description			
0..3	127:96	JIP Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td>S31</td></tr></table> The byte-aligned jump distance if a jump is taken for the channel.			S31
		S31			
95:64	UIP Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td>S31</td></tr></table> The byte aligned jump distance if a jump is taken for the instruction.			S31	
	S31				



brc - Branch Converging		
	63:32	Operand Control Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header Format: EU_INSTRUCTION_HEADER



Branch Diverging

brd - Branch Diverging			
Source:	Eulsa		
Length Bias:	4		
Description			
The brd instruction redirects the execution forward or backward to the instruction pointed by (current IP + offset). The jump will occur if any channels are branched away.			
In GEN binary, JIP is at location src1 when immediate and at location src0 when reg32, where reg32 is accessed as a scalar DWord. The ip register must be used (for example, by the assembler) as dst.			
Format: [[pred]] brd (exec_size) JIP			
Restriction			
Restriction : A brd instruction cannot use the Switch instruction option.			
Syntax			
[[pred]] brd (exec_size) imm32 [[pred]] brd (exec_size) reg32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < 32; n++) { if (WrEn[n]) { PcIP[n] = IP + JIP; } else { PcIP[n] = IP + 1; } } if (any PcIP == ExIP + JIP) { // any channel Jump(ExIP + JIP); }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types			
D			
DWord	Bit	Description	
0..3	127:96	JIP Format: S31 Jump Target Offset. The relative offset in bytes if a jump is taken for the instruction.	
	95	Source 0 Address Immediate [9] Sign Bit	
	94:91	Src1.SrcType Format: SrcType	
	90:89	Src1.RegFile Format: RegFile	



brd - Branch Diverging	
88:64	Source 0
	Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16')
	Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
88:64	Source 0
	Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1')
	Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1
63:32	Operand Control
	Format: EU_INSTRUCTION_OPERAND_CONTROLS
31:0	Header
	Format: EU_INSTRUCTION_HEADER



Break

break - Break				
Source:	Eulsa			
Length Bias:	4			
Description				
<p>The break instruction is used to early-out from the inner most loop, or early out from the inner most switch block.</p> <p>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 offset to the while instruction of the loop block.</p> <p>If SPF is ON, the UIP must be used to update IP; JIP is not used in this case</p> <p>The following table describes the two 32-bit instruction pointer offsets. Both the JIP and UIP are signed 32-bit numbers, added to IP pre-increment. In GEN binary, JIP and UIP are at locations src0 and src1 and must be of type DW (signed DWord integer). When the offsets are immediate, src0 regfile must be immediate.</p>				
<p>Format:</p> <p>[(pred)] break (exec_size) JIP UIP</p>				
Syntax				
<p>[(pred)] break (exec_size) imm32 imm32</p>				
Pseudocode				
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.channel[n]) { PcIP[n] = IP + UIP; else { PcIP[n] = IP + 1; } } if (PcIP != (IP + 1)) { // all channels Jump(IP + JIP); }</pre>				
Predication	Conditional Modifier	Saturation	Source Modifier	
Y	N	N	N	
DWord	Bit	Description		
0.3	127:96	JIP		
		Format:	S31	
	The byte-aligned jump distance if a jump is taken for the channel.			
	95:64	UIP		
Format:		S31		
The byte aligned jump distance if a jump is taken for the instruction.				
63:32	Operand Control			
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
31:0	Header			
	Format:	EU_INSTRUCTION_HEADER		



Byte Scaled Read MSD

MSD2R_BS - Byte Scaled Read MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Scaled R/W	
Group:	Byte Scaled R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_A32_MHP If present, modifies the address calculations.
	18:15	Message Type Default Value: 04h Format: Opcode Byte Scattered Read message
	14:12	Reserved Format: MBZ Ignored
	11:10	Data Elements Format: MDC_DS Specifies the number of Bytes to be read or written per Dword
	9	Reserved Format: MBZ Ignored
	8	SIMD Mode Format: MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)
	7:0	Sideband Scaled Offset Format: MDC_A32_SBSO In combination with Header Present field, specifies the Scale pitch and the Offset for the message.



Byte Scaled Write MSD

MSD2W_BS - Byte Scaled Write MSD						
Source:	DataPort 2					
Length Bias:	1					
Family:	Scaled R/W					
Group:	Byte Scaled R/W					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_A32_MHP</td> </tr> </table> <p>If present, modifies the address calculations.</p>	Format:	MDC_A32_MHP		
	Format:	MDC_A32_MHP				
	18:15	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>0Ch</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Byte Scattered Write message</p>	Default Value:	0Ch	Format:	Opcode
	Default Value:	0Ch				
	Format:	Opcode				
	14:12	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
	Format:	MBZ				
11:10	<p>Data Elements</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_DS</td> </tr> </table> <p>Specifies the number of Bytes to be read or written per Dword</p>	Format:	MDC_DS			
Format:	MDC_DS					
9	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
8	<p>SIMD Mode</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_SM2</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2			
Format:	MDC_SM2					
7:0	<p>Sideband Scaled Offset</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_A32_SBSO</td> </tr> </table> <p>In combination with Header Present field, specifies the Scale pitch and the Offset for the message.</p>	Format:	MDC_A32_SBSO			
Format:	MDC_A32_SBSO					



Byte Scattered Read MSD

MSDOR_BS - Byte Scattered Read MSD		
Source:	DataPort 0	
Length Bias:	1	
Family:	Scattered R/W	
Group:	Byte Scattered R/W	
DWord	Bit	Description
0	19	Header Present
		Format: Enable If set, indicates that the message includes the header.
	18	Legacy Message
		Default Value: 0h
		Format: Opcode Legacy Message
	17:14	Message Type
		Default Value: 04h
		Format: Opcode Byte Scattered Read message
13	Reserved	
	Format: MBZ Ignored	
12	Reserved	
	Format: MBZ Ignored	
11:10	Data Elements	
	Format: MDC_DS Specifies the number of Bytes to be read or written per Dword	
9	Reserved	
	Format: MBZ Ignored	



MSDOR_BS - Byte Scattered Read MSD			
8	<p>SIMD Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_SM2</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2
Format:	MDC_SM2		
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_SLM_A32
Format:	MDC_BTS_SLM_A32		



Byte Scattered Write MSD

MSD0W_BS - Byte Scattered Write MSD		
Source:	DataPort 0	
Length Bias:	1	
Family:	Scattered R/W	
Group:	Byte Scattered R/W	
DWord	Bit	Description
0	19	Header Present Format: <input type="checkbox"/> Enable If set, indicates that the message includes the header.
	18	Legacy Message Default Value: <input type="checkbox"/> 0h Format: <input type="checkbox"/> Opcode Legacy Message
	17:14	Message Type Default Value: <input type="checkbox"/> 0Ch Format: <input type="checkbox"/> Opcode Byte Scattered Write message
	13:12	Reserved Format: <input type="checkbox"/> MBZ Ignored
	11:10	Data Elements Format: <input type="checkbox"/> MDC_DS Specifies the number of Bytes to be read or written per Dword
	9	Reserved Format: <input type="checkbox"/> MBZ Ignored
	8	SIMD Mode Format: <input type="checkbox"/> MDC_SM2 Specifies the SIMD mode of the message (number of slots processed)



MSD0W_BS - Byte Scattered Write MSD			
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_SLM_A32
Format:	MDC_BTS_SLM_A32		



Call

call - Call			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>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.</p> <p>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.</p> <p>When SPF is on, the predication control must be scalar.</p> <p>The following section describes JIP, the jump offset.</p> <p>JIP can be an immediate or register value. When a jump occurs, this value is added to IP pre-increment. In GEN binary, JIP is at location src1 and src0 must be null. The GRF register must be put (for example, by the assembler) at dst location.</p> <p>Format: [(pred)] call (exec_size) dst JIP</p> <p>Format: [(pred)] call (exec_size) dst JIP</p>			
Restriction			
Restriction : The call instruction must have DWord source and destination type, and the destination must be QWord aligned.			
Restriction : A call instruction must target an instruction with Switch set; in addition, the instruction following the call must also have Switch set.			
Syntax			
[(pred)] call (exec_size) reg imm32 [(pred)] call (exec_size) reg reg32			
Pseudocode			
<pre>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); }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Dst Types			
D, UD			
DWord	Bit	Description	
0..3	127:96	JIP	



call - Call							
	<table border="1"> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td colspan="2">Jump Target Offset. The relative offset in bytes if a jump is taken for the instruction.</td> </tr> </table>	Format:	S31	Jump Target Offset. The relative offset in bytes if a jump is taken for the instruction.			
Format:	S31						
Jump Target Offset. The relative offset in bytes if a jump is taken for the instruction.							
95	Source 0 Address Immediate [9] Sign Bit						
94:91	<table border="1"> <tr> <td colspan="2">Src1.SrcType</td> </tr> <tr> <td>Format:</td> <td>SrcType</td> </tr> </table>	Src1.SrcType		Format:	SrcType		
Src1.SrcType							
Format:	SrcType						
90:89	<table border="1"> <tr> <td colspan="2">Src1.RegFile</td> </tr> <tr> <td>Format:</td> <td>RegFile</td> </tr> </table>	Src1.RegFile		Format:	RegFile		
Src1.RegFile							
Format:	RegFile						
88:64	<table border="1"> <tr> <td colspan="2">Source 0</td> </tr> <tr> <td>Exists If:</td> <td>(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode] == 'Align16')</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16</td> </tr> </table>	Source 0		Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode] == 'Align16')	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
Source 0							
Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode] == 'Align16')						
Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16						
88:64	<table border="1"> <tr> <td colspan="2">Source 0</td> </tr> <tr> <td>Exists If:</td> <td>(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode] == 'Align1')</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1</td> </tr> </table>	Source 0		Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode] == 'Align1')	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1
Source 0							
Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode] == 'Align1')						
Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1						
63:32	<table border="1"> <tr> <td colspan="2">Operand Control</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_OPERAND_CONTROLS</td> </tr> </table>	Operand Control		Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
Operand Control							
Format:	EU_INSTRUCTION_OPERAND_CONTROLS						
31:0	<table border="1"> <tr> <td colspan="2">Header</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_HEADER</td> </tr> </table>	Header		Format:	EU_INSTRUCTION_HEADER		
Header							
Format:	EU_INSTRUCTION_HEADER						



Call Absolute

calla - Call Absolute			
Source:	Eulsa		
Length Bias:	4		
<p>The calla 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 calla instruction. If none of the channels jump into the subroutine, the calla instruction is treated as a nop.</p> <p>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.</p> <p>If SPF is ON, none of the PcIP are updated.</p> <p>When SPF is on, the predication control must be scalar.</p> <p>The difference between calla and call is that calla uses JIP as the IP value rather than adding it to the IP value.</p>			
Format: [[pred]] calla (exec_size) dst JIP			
Restriction			
Restriction : The calla instruction must have DWord source and destination type, and the destination must be QWord-aligned.			
Restriction : The src0 regioning control must be <2;2,1>.			
Syntax			
[[pred]] calla (exec_size) reg imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.channel[n]) { PcIP[n] = 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(JIP); }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Dst Types			
D, UD			
DWord	Bit	Description	
0..3	127:96	JIP	
		Format:	S31
Jump Target Offset. The relative offset in bytes if a jump is taken for the instruction.			
	95	Source 0 Address Immediate [9] Sign Bit	



calla - Call Absolute		
94:91	Src1.SrcType	
	Format:	SrcType
90:89	Src1.RegFile	
	Format:	RegFile
88:64	Source 0	
	Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16')
	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
88:64	Source 0	
	Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1')
	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1
63:32	Operand Control	
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS
31:0	Header	
	Format:	EU_INSTRUCTION_HEADER



Compare

cmp - Compare			
Source:	Eulsa		
Length Bias:	4		
<p>The <code>cmp</code> instruction performs component-wise comparison of <code>src0</code> and <code>src1</code> and stores the results in the selected flag register and in <code>dst</code>. It takes component-wise subtraction of <code>src0</code> and <code>src1</code>, 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 <code>dst</code> channels. If the <code>dst</code> 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.</p> <p>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 <code>.u</code> (unordered).</p> <p>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 0xFFFFFFFF) is assigned to <code>dst</code>.</p> <p>When any source type is floating-point, the <code>cmp</code> instruction obeys the rules described in the tables in the Floating Point Modes section of the Data Types chapter.</p>			
Format: <code>[(pred)] cmp[.cmod] (exec_size) dst src0 src1</code>			
Syntax			
<code>[(pred)] cmp[.cmod] (exec_size) reg reg reg [(pred)] cmp[.cmod] (exec_size) reg reg imm32</code>			
Pseudocode			
<pre>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>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	N	Y
Src Types		Dst Types	
*B,*W,*D		*B,*W,*D	
*B,*W,*D		F	
F		F	
DF		DF	
HF		HF	
*B,*W,*D		HF	



cmp - Compare		
*W,*D		*W,*D
*W,*D		*Q
*Q		*W,*D
*Q		*Q
HF, F		HF, F
DWord	Bit	Description
0..3	127:64	RegSource
		Exists If: ([RegSource][Src1.RegFile]!='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile]='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
31:0	Header	
	Format: EU_INSTRUCTION_HEADER	



Compare NaN

cmpn - Compare NaN				
Source:	Eulsa			
Length Bias:	4			
<p>The cmpn instruction performs component-wise special-NaN 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 signals including NS based on the conditional modifier, and storing the conditional flag 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.</p> <p>A conditional modifier must be specified; the conditional modifier field cannot be 0000b. More information about the conditional signals used is in the Creating Conditional Flags section.</p> <p>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 0xFFFFFFFF) is assigned to dst.</p> <p>Min/Max instructions use cmpn to select the destination from the input sources (see the Min Max of Floating Point Numbers section for details).</p>				
Format:	[(pred)] cmpn[.cmod] (exec_size) dst src0 src1			
Restriction				
Restriction :				
.l and .ge are the only two conditional modifiers are supported for this instruction.				
Syntax				
[(pred)] cmpn[.cmod] (exec_size) reg reg reg [(pred)] cmpn[.cmod] (exec_size) reg reg imm32				
Pseudocode				
<pre>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] = ConditionNaN(results[n]); dst.chan[n][0] = bitMask[n]; // All bits for dst channel } } flag# = bitMask;</pre>				
Predication	Conditional Modifier	Saturation	Source Modifier	
Y	Y	N	Y	
Src Types			Dst Types	
*B,*W,*D			*B,*W,*D	
*B,*W,*D			F	
F			F	



cmpn - Compare NaN			
DF		DF	
HF		HF	
F, HF		F, HF	
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	(([RegSource][Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG
	127:64	ImmSource	
		Exists If:	(([ImmSource][Src1.RegFile]='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_IMM
	63:32	Operand Controls	
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Conditional Select

csel - Conditional Select			
Source:	Eulsa		
Length Bias:	4		
<p>The csel instruction selectively moves components in src0 or src1 to the dst based on the result of the compare of src2 with zero. If the channel condition is true, data in src0 is moved into dst. Otherwise, data in src1 is moved into dst. The csel instruction provides the function of a cmp followed by sel. The instruction must not be used if cmpn is required. The instruction does not update the flag register.</p> <p>The comparison follows the same rule as cmp instruction for that data type.</p>			
Format:	csel (exec_size) dst src0 src1 src2		
Syntax			
csel[.cmod] (exec_size) reg reg reg reg			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { bitMask[n] = 0; if (EMask.chan[n]) { result[n] = src2.chan[n] - 0; bitMask[n] = Condition(result[n]); if (bitMask[n] = 1) { dst.chan[n] = src0.chan[n]; } else { dst.chan[n] = src1.chan[n]; } } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
N	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:126	Reserved Format: MBZ	
	125:106	Source 2 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC	
	105	Reserved Format: MBZ	
	104:85	Source 1 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC	
	84	Reserved Format: MBZ	
	83:64	Source 0 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC	



csel - Conditional Select											
63:56	<p>Destination Register Number</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">DstRegNum</td> </tr> </table>	Format:	DstRegNum								
Format:	DstRegNum										
55:53	<p>Destination Subregister Number</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">DstSubRegNum[2:0]</td> </tr> </table>	Format:	DstSubRegNum[2:0]								
Format:	DstSubRegNum[2:0]										
52:49	<p>Destination Channel Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">ChanEn[4]</td> </tr> </table> <p>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</p>	Format:	ChanEn[4]								
Format:	ChanEn[4]										
48	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
47	<p>NibCtrl</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">NibCtrl</td> </tr> </table>	Format:	NibCtrl								
Format:	NibCtrl										
46	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
45:44	<p>Destination Data Type This field contains the data type for the destination</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e6f2ff;"> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Single Precision Float</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>DWord</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Unsigned DWord</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Double Precision Float</td> </tr> </tbody> </table>	Value	Name	00b	Single Precision Float	01b	DWord	10b	Unsigned DWord	11b	Double Precision Float
Value	Name										
00b	Single Precision Float										
01b	DWord										
10b	Unsigned DWord										
11b	Double Precision Float										
48:42	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
43:42	<p>Source Data Type This field contains the data type for all three sources</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e6f2ff;"> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Single Precision Float</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>DWord</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Unsigned DWord</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Double Precision Float</td> </tr> </tbody> </table>	Value	Name	00b	Single Precision Float	01b	DWord	10b	Unsigned DWord	11b	Double Precision Float
Value	Name										
00b	Single Precision Float										
01b	DWord										
10b	Unsigned DWord										
11b	Double Precision Float										



csel - Conditional Select		
41:40	Source 2 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
39:38	Source 1 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
41:36	Reserved	
	Exists If:	/// ([Property[Source Modifier] == 'false')
	Format:	MBZ
37:36	Source 0 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
35	Reserved	
	Format:	MBZ
34	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	Destination Register File	
	Value	Name
	0	GRF
	1	MRF
32	Reserved	
	Format:	MBZ
31:0	Header	
	Format:	EU_INSTRUCTION_HEADER



Conditional Send Message

sendc - Conditional Send Message			
Source:	Eulsa		
Length Bias:	4		
<p>The sendc instruction has the same behavior as the send instruction except the following.</p> <p>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.</p> <p>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.</p>			
Format:	[(pred)] sendc (exec_size) dst src0 exdesc desc		
Restriction			
Restriction : The sendc instruction has the same restrictions as the send instruction.			
Syntax			
[(pred)] sendc (exec_size) reg reg imm32 reg32a [(pred)] sendc (exec_size) reg reg imm32 imm32			
Pseudocode			
<pre>if (TDR[7] ... TDR[2] TDR[1] TDR[0]) { wait; } Evaluate(WrEn); MsgChEnable = WrEn; SourceReg = src0.RegNum; MessageEnqueue(MsgChEnable, ResponseReg, SourceReg, desc, exdesc);</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
DWord	Bit	Description	
0..3	127:96	Message	
		Format:	EU_INSTRUCTION_OPERAND_SEND_MSG
	95:89	Flags	
		Format:	EU_INSTRUCTION_FLAGS
	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[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16')	
Format:		EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16	



sendc - Conditional Send Message		
	63:32	Operand Control Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:28	Controls B Format: EU_INSTRUCTION_CONTROLS_B
	27:24	Shared Function ID (SFID) Format: SFID
	23:8	Controls A Format: EU_INSTRUCTION_CONTROLS_A
	7	Reserved Format: MBZ
	6:0	Opcode Format: EU_OPCODE



Conditional Split Send Message

sendsc - Conditional Split Send Message			
Source:	Eulsa		
Length Bias:	4		
<p>The sendsc instruction has the same behavior as the sends instruction except the following. sendsc first checks the dependent threads inside the Thread Dependency Register. There are up to 8 dependent threads in the TDR register. The sendsc instruction executes only when all the dependent threads in the TDR register are retired.</p> <p>Wait for dependencies in the TDR Register to clear, then send a message stored in GRF locations starting at <src0> followed by <src1> to a shared function identified by <ex_desc> along with control from <desc> and <ex_desc> with a GRF writeback location at <dest>.</p> <p>Format: [(pred)] sendsc (exec_size) <dest> <src0> <src1> <ex_desc> <desc></p>			
Restriction			
Restriction : The sendsc instruction has the same restrictions as the sends instruction.			
Syntax			
<pre>[(pred)] sendsc (exec_size) reg greg reg imm32 reg32a [(pred)] sendsc (exec_size) reg greg reg imm32 imm32 [(pred)] sendsc (exec_size) reg greg reg reg32a imm32 [(pred)] sendsc (exec_size) reg greg reg reg32a reg32a</pre>			
Pseudocode			
<pre>if (TDR[7] ... TDR[2] TDR[1] TDR[0]) { wait; } Evaluate(WrEn); <MsgChEnable> = WrEn; <SourceReg0> = <src0>.RegNum; <SourceReg1> = <src1>.RegNum; MessageEnqueue(<MsgChEnable>, <ResponseReg>, <SourceReg0>, <SourceReg1>, <ex_desc>, <dest>);</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
DWord	Bit	Description	
0..3	127:96	Message Format: EU_INSTRUCTION_OPERAND_SEND_MSG	
	95:80	ExDesc[31:16] Format: ExtMsgDescpt[31:16]	
	79	Source 0 Addressing Mode Format: AddrMode	



sendsc - Conditional Split Send Message		
78	Reserved	
	Exists If:	(([Source 0 Addressing Mode]='Direct')
	Format:	MBZ
78	Source 0 Address Immediate Sign [9]	
	Exists If:	(([Source 0 Addressing Mode]='Indirect')
	Format:	S9[9]
77	SelReg32Desc	
76:73	Source 0 Address Subregister Number	
	Exists If:	(([Source 0 Addressing Mode]='Indirect')
76:69	Source 0 Register Number	
	Exists If:	(([Source 0 Addressing Mode]='Direct')
72:68	Source 0 Address Immediate [8:4]	
	Exists If:	(([Source 0 Addressing Mode]='Indirect')
	Format:	S9[8:4]
68	Source 0 Subregister Number	
	Exists If:	(([Source 0 Addressing Mode]='Direct')
67:64	ExDesc[9:6]	
	Format:	ExtMsgDescpt[9:6]
63	Destination Addressing Mode	
	Format:	AddrMode
62	Destination Address Immediate Sign [9]	
	Exists If:	(([Destination Addressing Mode]='Indirect')
	Format:	S9[9]
62	Reserved	
	Exists If:	(([Destination Addressing Mode]='Direct')
	Format:	MBZ
61	Reserved	
	Format:	MBZ
60:57	Destination Address Subregister Number	
	Exists If:	(([Destination Addressing Mode]='Indirect')
60:53	Destination Register Number	
	Exists If:	(([Destination Addressing Mode]='Direct')
56:52	Destination Address Immediate [8:4]	
	Exists If:	(([Destination Addressing Mode]='Indirect')
	Format:	S9[8:4]



sendsc - Conditional Split Send Message		
52	Destination Subregister Number [4]	Exists If: <input type="checkbox"/> ([Destination Addressing Mode] == 'Direct')
51:44	Source 1 Register Number	
43:41	Reserved	Format: <input type="checkbox"/> MBZ
40:37	Destination Type	
36	Source 1 Register File	Format: <input type="checkbox"/> RegFile[0]
35	Destination Register File	Format: <input type="checkbox"/> RegFile[0]
34	MaskCtrl	
33:32	Flag Register Number/Subregister Number	
31:28	Controls B	Format: <input type="checkbox"/> EU_INSTRUCTION_CONTROLS_B
27:24	Shared Function ID (SFID)	Format: <input type="checkbox"/> SFID
23:8	Controls A	Format: <input type="checkbox"/> EU_INSTRUCTION_CONTROLS_A
7	Reserved	Format: <input type="checkbox"/> MBZ
6:0	Opcode	Format: <input type="checkbox"/> EU_OPCODE



Constant Cache Dword Scattered Read MSD

MSD_CC_DWS - Constant Cache Dword Scattered Read MSD		
Source:	Read-Only DataPort	
Length Bias:	1	
Family:	Scattered R/W	
Group:	DW Scattered R/W	
DWord	Bit	Description
0	19	Header Present
		Format: <input type="checkbox"/> Enable
	If set, indicates that the message includes the header.	
	18	Legacy Message
		Default Value: <input type="checkbox"/> 0h
		Format: <input type="checkbox"/> Opcode
	Legacy Message	
	17:14	Message Type
Default Value: <input type="checkbox"/> 03h		
Format: <input type="checkbox"/> Opcode		
Dword Scattered Read message		
13	Invalidate After Read	
	Format: <input type="checkbox"/> MDC_IAR	
Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs		
12	Reserved	
	Format: <input type="checkbox"/> MBZ	
Ignored		
11:10	Reserved	
	Format: <input type="checkbox"/> MBZ	
Ignored		
9	Legacy SIMD Mode	
	Default Value: <input type="checkbox"/> 1h	
	Format: <input type="checkbox"/> Opcode	
Must be set for compatibility.		



MSD_CC_DWS - Constant Cache Dword Scattered Read MSD			
8	<p>SIMD Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_SM2</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2
Format:	MDC_SM2		
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_BTS</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS
Format:	MDC_BTS		



Constant Cache Oword Block Read MSD

MSD_CC_OWB - Constant Cache Oword Block Read MSD		
Source:	Read-Only DataPort	
Length Bias:	1	
Family:	Block R/W	
Group:	OW Block R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header.
	18	Reserved Format: MBZ Ignored
	17:14	Message Type Default Value: 00h Format: Opcode Oword Block Read Constant Cache message
	13	Invalidate After Read Format: MDC_IAR Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs
	12:11	Reserved Format: MBZ Ignored
	10:8	Data Elements Format: MDC_DB_OW Specifies the number of contiguous Owords to be read or written
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Constant Cache Oword Dual Block Read MSD

MSD_CC_OWDB - Constant Cache Oword Dual Block Read MSD						
Source:	Read-Only DataPort					
Length Bias:	1					
Family:	Block R/W					
Group:	OW Dual Block R/W					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> If set, indicates that the message includes the header.	Format:	Enable		
	Format:	Enable				
	18	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ		
	Format:	MBZ				
	17:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>02h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Oword Block Read message	Default Value:	02h	Format:	Opcode
	Default Value:	02h				
	Format:	Opcode				
	13	Invalidate After Read <table border="1"> <tr> <td>Format:</td> <td>MDC_IAR</td> </tr> </table> Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs	Format:	MDC_IAR		
Format:	MDC_IAR					
12:10	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
9:8	Data Elements <table border="1"> <tr> <td>Format:</td> <td>MDC_DB_OWD</td> </tr> </table> Specifies the number of contiguous Owords to be read or written	Format:	MDC_DB_OWD			
Format:	MDC_DB_OWD					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS			
Format:	MDC_BTS					



Constant Cache Oword Unaligned Block Read MSD

MSD_CC_OWUB - Constant Cache Oword Unaligned Block Read MSD						
Source:	Read-Only DataPort					
Length Bias:	1					
Family:	Block R/W					
Group:	OW Unaligned Block R/W					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_MHR</td> </tr> </table> Indicates that the message requires a header.	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table> Ignored	Format:	MBZ		
	Format:	MBZ				
	17:14	Message Type <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Default Value:</td> <td style="text-align: center;">01h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Oword Unaligned Block Read Constant Cache message	Default Value:	01h	Format:	Opcode
	Default Value:	01h				
Format:	Opcode					
13:11	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
10:8	Data Elements <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_DB_OW</td> </tr> </table> Specifies the number of contiguous Owords to be read	Format:	MDC_DB_OW			
Format:	MDC_DB_OW					
7:0	Binding Table Index <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_BTS</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS			
Format:	MDC_BTS					



Continue

cont - Continue			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>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 instruction 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.</p> <p>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.</p> <p>If SPF is ON, the UIP must be used to update IP; JIP is not used in this case.</p> <p>The following table describes the two 32-bit instruction pointer offsets. Both the JIP and UIP are signed 32-bit numbers, added to IP pre-increment. In GEN binary, JIP and UIP are at locations src0 and src1 and must be of type DW (signed DWord integer). When the offsets are immediate, src0 regfile must be immediate.</p>			
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) imm32 imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.channel[n]) { if (PMask[n]) { // 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); }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
DWord	Bit	Description	
0..3	127:96	JIP Format: S31 The byte-aligned jump distance if a jump is taken for the channel.	



cont - Continue		
	95:64	UIP Format: S31 The byte aligned jump distance if a jump is taken for the instruction.
	63:32	Operand Control Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header Format: EU_INSTRUCTION_HEADER



Count Bits Set

cbit - Count Bits Set			
Source:	Eulsa		
Length Bias:	4		
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.			
Syntax			
[[pred]] cbit (exec_size) reg reg [[pred]] cbit (exec_size) reg imm32			
Pseudocode			
<pre>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; } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types	Dst Types		
UB, UW, UD	UD		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	[[Operand Controls][Src0.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource	
Exists If:		[[Operand Controls][Src0.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Dot Product 2

dp2 - Dot Product 2			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>The dot product of two vectors of equal length is the sum of the products of each pair of corresponding elements.</p> <p>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.</p>			
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			
<pre>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]; 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	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile] != 'IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG



dp2 - Dot Product 2		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
		Format: EU_INSTRUCTION_SOURCES_REG_IMM
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Dot Product 3

dp3 - Dot Product 3			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>The dot product of two vectors of equal length is the sum of the products of each pair of corresponding elements.</p> <p>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.</p>			
Format:	[(pred)] dp3[.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)] dp3[.cmod] (exec_size) reg reg reg [(pred)] dp3[.cmod] (exec_size) reg reg imm32			
Pseudocode			
<pre>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	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile] != 'IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG



dp3 - Dot Product 3		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
		Format: EU_INSTRUCTION_SOURCES_REG_IMM
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Dot Product 4

dp4 - Dot Product 4			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>The dot product of two vectors of equal length is the sum of the products of each pair of corresponding elements.</p>			
Format:	[(pred)] dp4[.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)] dp4[.cmod] (exec_size) reg reg reg [(pred)] dp4[.cmod] (exec_size) reg reg imm32			
Pseudocode			
<pre>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; }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	((RegSource)[Src1.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource	
Exists If:		((ImmSource)[Src1.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_REG_IMM		



dp4 - Dot Product 4		
	63:32	Operand Controls Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header Format: EU_INSTRUCTION_HEADER



Dot Product Homogeneous

dph - Dot Product Homogeneous			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p>			
Format:	[(pred)] dph[.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)] dph[.cmod] (exec_size) reg reg reg [(pred)] dph[.cmod] (exec_size) reg reg imm32			
Pseudocode			
<pre>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; }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource	
Exists If:		([ImmSource][Src1.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_REG_IMM		



dph - Dot Product Homogeneous		
	63:32	Operand Controls Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header Format: EU_INSTRUCTION_HEADER



Dword Atomic Counter Binary with Return Data Operation MSD

MSD1R_DWAC2 - Dword Atomic Counter Binary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Atomic Counter Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header
	18:14	Message Type Default Value: 0Bh Format: Opcode Atomic Counter Operation message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	SIMD Mode Format: MDC_SM2RS Specifies the SIMD mode of the message (number of slots processed)
	11:8	Atomic Integer Operation Format: MDC_AOP2 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword Atomic Counter Binary Write Only Operation MSD

MSD1W_DWAC2 - Dword Atomic Counter Binary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Atomic Counter Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_MHR</td> </tr> </table> Indicates that the message requires a header	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%; text-align: center;">0Bh</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Atomic Counter Operation message	Default Value:	0Bh	Format:	Opcode
	Default Value:	0Bh				
	Format:	Opcode				
	13	Return Data Control <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%; text-align: center;">0h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	SIMD Mode <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_SM2RS</td> </tr> </table> Specifies the SIMD mode of the message (number of slots processed)	Format:	MDC_SM2RS			
Format:	MDC_SM2RS					
11:8	Atomic Integer Operation <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_AOP2</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_BTS</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS			
Format:	MDC_BTS					



Dword Atomic Counter Unary with Return Data Operation MSD

MSD1R_DWAC1 - Dword Atomic Counter Unary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Atomic Counter Unary Operation					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHR</td> </tr> </table> <p>Indicates that the message requires a header</p>	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18:14	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>0Bh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Atomic Counter Operation message</p>	Default Value:	0Bh	Format:	Opcode
	Default Value:	0Bh				
	Format:	Opcode				
	13	<p>Return Data Control</p> <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that return data is sent back to the thread.</p>	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	<p>SIMD Mode</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_SM2RS</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2RS			
Format:	MDC_SM2RS					
11:8	<p>Atomic Integer Operation</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP1</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS			
Format:	MDC_BTS					



Dword Atomic Counter Unary Write Only Operation MSD

MSD1W_DWAC1 - Dword Atomic Counter Unary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Atomic Counter Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">MDC_MHR</td> </tr> </table> Indicates that the message requires a header	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%;">0Bh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Atomic Counter Operation message	Default Value:	0Bh	Format:	Opcode
	Default Value:	0Bh				
	Format:	Opcode				
	13	Return Data Control <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%;">0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	SIMD Mode <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">MDC_SM2RS</td> </tr> </table> Specifies the SIMD mode of the message (number of slots processed)	Format:	MDC_SM2RS			
Format:	MDC_SM2RS					
11:8	Atomic Integer Operation <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">MDC_AOP1</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">MDC_BTS</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS			
Format:	MDC_BTS					



Dword Scattered Read MSD

MSD0R_DWS - Dword Scattered Read MSD						
Source:	DataPort 0					
Length Bias:	1					
Family:	Scattered R/W					
Group:	DW Scattered R/W					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	Enable		
	Format:	Enable				
	18	Legacy Message <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Legacy Message</p>	Default Value:	0h	Format:	Opcode
	Default Value:	0h				
	Format:	Opcode				
	17:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>03h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Dword Scattered Read message</p>	Default Value:	03h	Format:	Opcode
	Default Value:	03h				
Format:	Opcode					
13	Invalidate After Read <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="color: red;">MDC_IAR</td> </tr> </table> <p>Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs</p>	Format:	MDC_IAR			
Format:	MDC_IAR					
12	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
11:10	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
9	Legacy SIMD Mode <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Must be set for compatibility.</p>	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					



MSD0R_DWS - Dword Scattered Read MSD			
8	<p>SIMD Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_SM2</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2
Format:	MDC_SM2		
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_BTS_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_A32
Format:	MDC_BTS_A32		



Dword Scattered Write MSD

MSD0W_DWS - Dword Scattered Write MSD						
Source:	DataPort 0					
Length Bias:	1					
Family:	Scattered R/W					
Group:	DW Scattered R/W					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	Enable		
	Format:	Enable				
	18	Legacy Message <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Legacy Message</p>	Default Value:	0h	Format:	Opcode
	Default Value:	0h				
	Format:	Opcode				
	17:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0Bh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Dword Scattered Write message</p>	Default Value:	0Bh	Format:	Opcode
	Default Value:	0Bh				
	Format:	Opcode				
13:12	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
11:10	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
9	Legacy SIMD Mode <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Must be set for compatibility.</p>	Default Value:	1h	Format:	Opcode	
Default Value:	1h					
Format:	Opcode					
8	SIMD Mode <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MDC_SM2</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2			
Format:	MDC_SM2					



MSD0W_DWS - Dword Scattered Write MSD			
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_BTS_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_A32
Format:	MDC_BTS_A32		



Dword SIMD4x2 Atomic Counter Binary with Return Data Operation MSD

MSD1R_DWAC2_4x2 - Dword SIMD4x2 Atomic Counter Binary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Atomic Counter Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHR</td> </tr> </table> Indicates that the message requires a header	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>0Ch</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Atomic Counter Operation SIMD4x2 message	Default Value:	0Ch	Format:	Opcode
	Default Value:	0Ch				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that return data is sent back to the thread.	Default Value:	1h	Format:	Opcode
	Default Value:	1h				
Format:	Opcode					
12	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS			
Format:	MDC_BTS					



Dword SIMD4x2 Atomic Counter Binary Write Only Operation MSD

MSD1W_DWAC2_4x2 - Dword SIMD4x2 Atomic Counter Binary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Atomic Counter Binary Operation	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHR Indicates that the message requires a header
	18:14	Message Type
		Default Value: 0Ch
		Format: Opcode Atomic Counter Operation SIMD4x2 message
	13	Return Data Control
Default Value: 0h		
Format: Opcode Specifies that no return data is sent back to the thread.		
12	Reserved	
	Format: MBZ Ignored	
11:8	Atomic Integer Operation	
	Format: MDC_AOP2 Specifies the atomic integer operation to be performed.	
7:0	Binding Table Index	
	Format: MDC_BTS Specifies the Binding Table Index for the message	



Dword SIMD4x2 Atomic Counter Unary with Return Data Operation MSD

MSD1R_DWAC1_4x2 - Dword SIMD4x2 Atomic Counter Unary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Atomic Counter Unary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header
	18:14	Message Type Default Value: 0Ch Format: Opcode Atomic Counter Operation SIMD4x2 message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP1 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword SIMD4x2 Atomic Counter Unary Write Only Operation MSD

MSD1W_DWAC1_4x2 - Dword SIMD4x2 Atomic Counter Unary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Atomic Counter Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHR</td> </tr> </table> Indicates that the message requires a header	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>0Ch</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Atomic Counter Operation SIMD4x2 message	Default Value:	0Ch	Format:	Opcode
	Default Value:	0Ch				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP1</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS			
Format:	MDC_BTS					



Dword SIMD4x2 Typed Atomic Integer Binary with Return Data Operation MSD

MSD1R_DWTAI2_4x2 - Dword SIMD4x2 Typed Atomic Integer Binary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Atomic Operation	
Group:	Dword Typed Atomic Integer Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 07h Format: Opcode Typed Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP2 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword SIMD4x2 Typed Atomic Integer Binary Write Only Operation MSD

MSD1W_DWTAI2_4x2 - Dword SIMD4x2 Typed Atomic Integer Binary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Atomic Operation	
Group:	Dword Typed Atomic Integer Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 07h Format: Opcode Typed Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP2 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword SIMD4x2 Typed Atomic Integer Trinary with Return Data Operation MSD

MSD1R_DWTAI3_4x2 - Dword SIMD4x2 Typed Atomic Integer Trinary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Atomic Operation	
Group:	Dword Typed Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 07h Format: Opcode Typed Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP3S Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword SIMD4x2 Typed Atomic Integer Trinary Write Only Operation MSD

MSD1W_DWTAI3_4x2 - Dword SIMD4x2 Typed Atomic Integer Trinary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Atomic Operation	
Group:	Dword Typed Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 07h Format: Opcode Typed Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP3S Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword SIMD4x2 Typed Atomic Integer Unary with Return Data Operation MSD

MSD1R_DWTAI1_4x2 - Dword SIMD4x2 Typed Atomic Integer Unary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Atomic Operation	
Group:	Dword Typed Atomic Integer Unary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 07h Format: Opcode Typed Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP1 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword SIMD4x2 Typed Atomic Integer Unary Write Only Operation MSD

MSD1W_DWTAI1_4x2 - Dword SIMD4x2 Typed Atomic Integer Unary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Atomic Operation	
Group:	Dword Typed Atomic Integer Unary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 07h Format: Opcode Typed Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP1 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword SIMD4x2 Untyped Atomic Float Binary with Return Data Operation MSD

MSD1R_DWAF2_4x2 - Dword SIMD4x2 Untyped Atomic Float Binary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Float Binary Operation	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type
		Default Value: 1Ch
		Format: Opcode Untyped Atomic Float Operation SIMD4x2 message
	13	Return Data Control
Default Value: 1h		
Format: Opcode Specifies that return data is sent back to the thread.		
12	Reserved	
	Format: MBZ Ignored	
11	Data Width	
	Default Value: 0h	
	Format: Opcode Operations are on 32-bit floats.	
10	Reserved	
	Format: MBZ Ignored	



MSD1R_DWAF2_4x2 - Dword SIMD4x2 Untyped Atomic Float Binary with Return Data Operation MSD									
	<table border="1"> <tr> <td style="text-align: center; vertical-align: top;">9:8</td> <td> <p>Atomic Float Operation</p> <table border="1"> <tr> <td>Format:</td> <td style="text-align: center;">MDC_FOP2</td> </tr> </table> <p>Specifies the atomic float operation to be performed.</p> </td> </tr> <tr> <td style="text-align: center; vertical-align: top;">7:0</td> <td> <p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td style="text-align: center;">MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p> </td> </tr> </table>	9:8	<p>Atomic Float Operation</p> <table border="1"> <tr> <td>Format:</td> <td style="text-align: center;">MDC_FOP2</td> </tr> </table> <p>Specifies the atomic float operation to be performed.</p>	Format:	MDC_FOP2	7:0	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td style="text-align: center;">MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_SLM_A32
	9:8	<p>Atomic Float Operation</p> <table border="1"> <tr> <td>Format:</td> <td style="text-align: center;">MDC_FOP2</td> </tr> </table> <p>Specifies the atomic float operation to be performed.</p>	Format:	MDC_FOP2					
Format:	MDC_FOP2								
7:0	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td style="text-align: center;">MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_SLM_A32						
Format:	MDC_BTS_SLM_A32								



Dword SIMD4x2 Untyped Atomic Float Binary Write Only Operation MSD

MSD1W_DWAF2_4x2 - Dword SIMD4x2 Untyped Atomic Float Binary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Float Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> If set, indicates that the message includes the header.	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>1Ch</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Untyped Atomic Float Operation SIMD4x2 message	Default Value:	1Ch	Format:	Opcode
	Default Value:	1Ch				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
11	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Operations are on 32-bit floats.	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
10	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					



MSD1W_DWAF2_4x2 - Dword SIMD4x2 Untyped Atomic Float Binary Write Only Operation MSD

	9:8	Atomic Float Operation Format: MDC_FOP2 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword SIMD4x2 Untyped Atomic Float Trinary with Return Data Operation MSD

MSD1R_DWAF3_4x2 - Dword SIMD4x2 Untyped Atomic Float Trinary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Float Trinary Operation					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>1Ch</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Untyped Atomic Float Operation SIMD4x2 message</p>	Default Value:	1Ch	Format:	Opcode
	Default Value:	1Ch				
	Format:	Opcode				
	13	<p>Return Data Control</p> <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that return data is sent back to the thread.</p>	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
11	<p>Data Width</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Operations are on 32-bit floats.</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
10	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					



MSD1R_DWAF3_4x2 - Dword SIMD4x2 Untyped Atomic Float Trinary with Return Data Operation MSD

	9:8	Atomic Float Operation	
		Format:	MDC_FOP3
		Specifies the atomic float operation to be performed.	
	7:0	Binding Table Index	
		Format:	MDC_BTS_SLM_A32
		Specifies the Binding Table Index for the message	



Dword SIMD4x2 Untyped Atomic Float Trinary Write Only Operation MSD

MSD1W_DWAF3_4x2 - Dword SIMD4x2 Untyped Atomic Float Trinary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Float Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 1Ch Format: Opcode Untyped Atomic Float Operation SIMD4x2 message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Reserved Format: MBZ Specifies the SIMD mode of the message (number of slots processed)
	11	Data Width Default Value: 0h Format: Opcode Operations are on 32-bit floats.
	10	Reserved Format: MBZ Ignored



MSD1W_DWAF3_4x2 - Dword SIMD4x2 Untyped Atomic Float Trinary Write Only Operation MSD

	9:8	Atomic Float Operation Format: MDC_FOP3 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword SIMD4x2 Untyped Atomic Integer Binary with Return Data Operation MSD

MSD1R_DWAI2_4x2 - Dword SIMD4x2 Untyped Atomic Integer Binary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> If set, indicates that the message includes the header.	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>03h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Untyped Atomic Integer Operation SIMD4x2 message	Default Value:	03h	Format:	Opcode
	Default Value:	03h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that return data is sent back to the thread.	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS_SLM_A32</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS_SLM_A32			
Format:	MDC_BTS_SLM_A32					



Dword SIMD4x2 Untyped Atomic Integer Binary Write Only Operation MSD

MSD1W_DWAI2_4x2 - Dword SIMD4x2 Untyped Atomic Integer Binary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 03h Format: Opcode Untyped Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP2 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword SIMD4x2 Untyped Atomic Integer Trinary with Return Data Operation MSD

MSD1R_DWAI3_4x2 - Dword SIMD4x2 Untyped Atomic Integer Trinary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 03h Format: Opcode Untyped Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP3 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword SIMD4x2 Untyped Atomic Integer Trinary Write Only Operation MSD

MSD1W_DWAI3_4x2 - Dword SIMD4x2 Untyped Atomic Integer Trinary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 03h Format: Opcode Untyped Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP3 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword SIMD4x2 Untyped Atomic Integer Unary with Return Data Operation MSD

MSD1R_DWAI1_4x2 - Dword SIMD4x2 Untyped Atomic Integer Unary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Unary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 03h Format: Opcode Untyped Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP1 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword SIMD4x2 Untyped Atomic Integer Unary Write Only Operation MSD

MSD1W_DWAI1_4x2 - Dword SIMD4x2 Untyped Atomic Integer Unary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Unary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 03h Format: Opcode Untyped Atomic Integer Operation SIMD4x2 message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	Reserved Format: MBZ Ignored
	11:8	Atomic Integer Operation Format: MDC_AOP1 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword Typed Atomic Integer Binary with Return Data Operation MSD

MSD1R_DWTAI2 - Dword Typed Atomic Integer Binary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Atomic Operation	
Group:	Dword Typed Atomic Integer Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 06h Format: Opcode Typed Atomic Integer Operation message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Slot Group Format: MDC_SG2 Specifies the Slot Group mode of the message (which slots are processed)
	11:8	Atomic Integer Operation Format: MDC_AOP2 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword Typed Atomic Integer Binary Write Only Operation MSD

MSD1W_DWTAI2 - Dword Typed Atomic Integer Binary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Typed Atomic Operation					
Group:	Dword Typed Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> If set, indicates that the message includes the header.	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>06h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Typed Atomic Integer Operation message	Default Value:	06h	Format:	Opcode
	Default Value:	06h				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Slot Group <table border="1"> <tr> <td>Format:</td> <td>MDC_SG2</td> </tr> </table> Specifies the Slot Group mode of the message (which slots are processed)	Format:	MDC_SG2			
Format:	MDC_SG2					
11:8	Atomic Integer Operation <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP2</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS			
Format:	MDC_BTS					



Dword Typed Atomic Integer Trinary with Return Data Operation MSD

MSD1R_DWTAI3 - Dword Typed Atomic Integer Trinary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Typed Atomic Operation					
Group:	Dword Typed Atomic Integer Trinary Operation					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">06h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Typed Atomic Integer Operation message</p>	Default Value:	06h	Format:	Opcode
	Default Value:	06h				
	Format:	Opcode				
	13	<p>Return Data Control</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">1h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Specifies that return data is sent back to the thread.</p>	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	<p>Slot Group</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_SG2</td> </tr> </table> <p>Specifies the Slot Group mode of the message (which slots are processed)</p>	Format:	MDC_SG2			
Format:	MDC_SG2					
11:8	<p>Atomic Integer Operation</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_AOP3S</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP3S			
Format:	MDC_AOP3S					
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_BTS</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS			
Format:	MDC_BTS					



Dword Typed Atomic Integer Trinary Write Only Operation MSD

MSD1W_DWTAI3 - Dword Typed Atomic Integer Trinary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Typed Atomic Operation					
Group:	Dword Typed Atomic Integer Trinary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_MHP</td> </tr> </table> If set, indicates that the message includes the header.	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	Message Type <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%; text-align: center;">06h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Typed Atomic Integer Operation message	Default Value:	06h	Format:	Opcode
	Default Value:	06h				
	Format:	Opcode				
	13	Return Data Control <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%; text-align: center;">0h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	Slot Group <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_SG2</td> </tr> </table> Specifies the Slot Group mode of the message (which slots are processed)	Format:	MDC_SG2			
Format:	MDC_SG2					
11:8	Atomic Integer Operation <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_AOP3S</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP3S			
Format:	MDC_AOP3S					
7:0	Binding Table Index <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_BTS</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS			
Format:	MDC_BTS					



Dword Typed Atomic Integer Unary with Return Data Operation MSD

MSD1R_DWTAI1 - Dword Typed Atomic Integer Unary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Atomic Operation	
Group:	Dword Typed Atomic Integer Unary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 06h Format: Opcode Typed Atomic Integer Operation message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	Slot Group Format: MDC_SG2 Specifies the Slot Group mode of the message (which slots are processed)
	11:8	Atomic Integer Operation Format: MDC_AOP1 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



Dword Typed Atomic Integer Unary Write Only Operation MSD

MSD1W_DWTAI1 - Dword Typed Atomic Integer Unary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Typed Atomic Operation					
Group:	Dword Typed Atomic Integer Unary Operation					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>06h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Typed Atomic Integer Operation message</p>	Default Value:	06h	Format:	Opcode
	Default Value:	06h				
	Format:	Opcode				
	13	<p>Return Data Control</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	<p>Slot Group</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_SG2</td> </tr> </table> <p>Specifies the Slot Group mode of the message (which slots are processed)</p>	Format:	MDC_SG2			
Format:	MDC_SG2					
11:8	<p>Atomic Integer Operation</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_AOP1</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS			
Format:	MDC_BTS					



Dword Untyped Atomic Float Binary with Return Data Operation MSD

MSD1R_DWAF2 - Dword Untyped Atomic Float Binary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Float Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> If set, indicates that the message includes the header.	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	Message Type <table border="1"> <tr> <td>Default Value:</td> <td>1Bh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Untyped Atomic Float Operation message	Default Value:	1Bh	Format:	Opcode
	Default Value:	1Bh				
	Format:	Opcode				
	13	Return Data Control <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Specifies that return data is sent back to the thread.	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	SIMD Mode <table border="1"> <tr> <td>Format:</td> <td>MDC_SM2R</td> </tr> </table> Specifies the SIMD mode of the message (number of slots processed)	Format:	MDC_SM2R			
Format:	MDC_SM2R					
11	Data Width <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Operations are on 32-bit floats.	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
10	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					



MSD1R_DWAF2 - Dword Untyped Atomic Float Binary with Return Data Operation MSD

	9:8	Atomic Float Operation	
		Format:	MDC_FOP2
		Specifies the atomic float operation to be performed.	
	7:0	Binding Table Index	
		Format:	MDC_BTS_SLM_A32
		Specifies the Binding Table Index for the message	



Dword Untyped Atomic Float Binary Write Only Operation MSD

MSD1W_DWAF2 - Dword Untyped Atomic Float Binary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Float Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 1Bh Format: Opcode Untyped Atomic Float Operation message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	SIMD Mode Format: MDC_SM2R Specifies the SIMD mode of the message (number of slots processed)
	11	Data Width Default Value: 0h Format: Opcode Operations are on 32-bit floats.
	10	Reserved Format: MBZ Ignored



MSD1W_DWAF2 - Dword Untyped Atomic Float Binary Write Only Operation MSD

	9:8	Atomic Float Operation Format: MDC_FOP2 Specifies the atomic float operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword Untyped Atomic Float Trinary with Return Data Operation MSD

MSD1R_DWAF3 - Dword Untyped Atomic Float Trinary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Float Trinary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	Message Type <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Default Value:</td> <td style="text-align: center;">1Bh</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Untyped Atomic Float Operation message</p>	Default Value:	1Bh	Format:	Opcode
	Default Value:	1Bh				
	Format:	Opcode				
	13	Return Data Control <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Default Value:</td> <td style="text-align: center;">1h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Specifies that return data is sent back to the thread.</p>	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	SIMD Mode <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_SM2R</td> </tr> </table> <p>Only SIMD8 operations are supported.</p>	Format:	MDC_SM2R			
Format:	MDC_SM2R					
11	Data Width <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Default Value:</td> <td style="text-align: center;">0h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Operations are on 32-bit floats.</p>	Default Value:	0h	Format:	Opcode	
Default Value:	0h					
Format:	Opcode					
10	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					



MSD1R_DWAF3 - Dword Untyped Atomic Float Trinary with Return Data Operation MSD				
	9:8	<p>Atomic Float Operation</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_FOP3</td> </tr> </table> <p>Specifies the atomic float operation to be performed.</p>	Format:	MDC_FOP3
	Format:	MDC_FOP3		
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_SLM_A32	
Format:	MDC_BTS_SLM_A32			



Dword Untyped Atomic Float Trinary Write Only Operation MSD

MSD1W_DWAF3 - Dword Untyped Atomic Float Trinary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Float Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 1Bh Format: Opcode Untyped Atomic Float Operation message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	SIMD Mode Format: MDC_SM2R Specifies the SIMD mode of the message (number of slots processed)
	11	Data Width Default Value: 0h Format: Opcode Operations are on 32-bit floats.
	10	Reserved Format: MBZ Ignored



MSD1W_DWAF3 - Dword Untyped Atomic Float Trinary Write Only Operation MSD

	9:8	Atomic Float Operation	
		Format:	MDC_FOP3
		Specifies the atomic float operation to be performed.	
	7:0	Binding Table Index	
		Format:	MDC_BTS_SLM_A32
		Specifies the Binding Table Index for the message	



Dword Untyped Atomic Integer Binary with Return Data Operation MSD

MSD1R_DWAI2 - Dword Untyped Atomic Integer Binary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Binary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">02h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Untyped Atomic Integer Operation message</p>	Default Value:	02h	Format:	Opcode
	Default Value:	02h				
	Format:	Opcode				
	13	Return Data Control <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">1h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Specifies that return data is sent back to the thread.</p>	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	SIMD Mode <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_SM2R</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2R			
Format:	MDC_SM2R					
11:8	Atomic Integer Operation <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_AOP2</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP2			
Format:	MDC_AOP2					
7:0	Binding Table Index <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_SLM_A32			
Format:	MDC_BTS_SLM_A32					



Dword Untyped Atomic Integer Binary Write Only Operation MSD

MSD1W_DWAI2 - Dword Untyped Atomic Integer Binary Write Only Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Binary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 02h Format: Opcode Untyped Atomic Integer Operation message
	13	Return Data Control Default Value: 0h Format: Opcode Specifies that no return data is sent back to the thread.
	12	SIMD Mode Format: MDC_SM2R Specifies the SIMD mode of the message (number of slots processed)
	11:8	Atomic Integer Operation Format: MDC_AOP2 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword Untyped Atomic Integer Trinary with Return Data Operation MSD

MSD1R_DWAI3 - Dword Untyped Atomic Integer Trinary with Return Data Operation MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Atomic Operation	
Group:	Dword Untyped Atomic Integer Trinary Operation	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type Default Value: 02h Format: Opcode Untyped Atomic Integer Operation message
	13	Return Data Control Default Value: 1h Format: Opcode Specifies that return data is sent back to the thread.
	12	SIMD Mode Format: MDC_SM2R Specifies the SIMD mode of the message (number of slots processed)
	11:8	Atomic Integer Operation Format: MDC_AOP3 Specifies the atomic integer operation to be performed.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Dword Untyped Atomic Integer Trinary Write Only Operation MSD

MSD1W_DWAI3 - Dword Untyped Atomic Integer Trinary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Trinary Operation					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td style="text-align: center;">02h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Untyped Atomic Integer Operation message</p>	Default Value:	02h	Format:	Opcode
	Default Value:	02h				
	Format:	Opcode				
	13	<p>Return Data Control</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td style="text-align: center;">0h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Specifies that no return data is sent back to the thread.</p>	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	<p>SIMD Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_SM2R</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2R			
Format:	MDC_SM2R					
11:8	<p>Atomic Integer Operation</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_AOP3</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP3			
Format:	MDC_AOP3					
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_SLM_A32			
Format:	MDC_BTS_SLM_A32					



Dword Untyped Atomic Integer Unary with Return Data Operation MSD

MSD1R_DWAI1 - Dword Untyped Atomic Integer Unary with Return Data Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">02h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Untyped Atomic Integer Operation message</p>	Default Value:	02h	Format:	Opcode
	Default Value:	02h				
	Format:	Opcode				
	13	Return Data Control <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">1h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> <p>Specifies that return data is sent back to the thread.</p>	Default Value:	1h	Format:	Opcode
Default Value:	1h					
Format:	Opcode					
12	SIMD Mode <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_SM2R</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM2R			
Format:	MDC_SM2R					
11:8	Atomic Integer Operation <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_AOP1</td> </tr> </table> <p>Specifies the atomic integer operation to be performed.</p>	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_SLM_A32			
Format:	MDC_BTS_SLM_A32					



Dword Untyped Atomic Integer Unary Write Only Operation MSD

MSD1W_DWAI1 - Dword Untyped Atomic Integer Unary Write Only Operation MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Atomic Operation					
Group:	Dword Untyped Atomic Integer Unary Operation					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_MHP</td> </tr> </table> If set, indicates that the message includes the header.	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">02h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Untyped Atomic Integer Operation message	Default Value:	02h	Format:	Opcode
	Default Value:	02h				
	Format:	Opcode				
	13	Return Data Control <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">0h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Specifies that no return data is sent back to the thread.	Default Value:	0h	Format:	Opcode
Default Value:	0h					
Format:	Opcode					
12	SIMD Mode <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_SM2R</td> </tr> </table> Specifies the SIMD mode of the message (number of slots processed)	Format:	MDC_SM2R			
Format:	MDC_SM2R					
11:8	Atomic Integer Operation <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_AOP1</td> </tr> </table> Specifies the atomic integer operation to be performed.	Format:	MDC_AOP1			
Format:	MDC_AOP1					
7:0	Binding Table Index <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_BTS_SLM_A32</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS_SLM_A32			
Format:	MDC_BTS_SLM_A32					



Else

else - Else			
Source:	Eulsa		
Length Bias:	4		
<p>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 prior to entering the if/endif block remain inactive throughout the entire block.</p> <p>All enabled channels upon arriving the else instruction will be redirected to the matching endif. If all channels are redirected (by else or before else), a relative jump is performed to the location specified by <JIP>. The jump target should be the matching endif instruction for that conditional block.</p> <p>The following table describes the 32-bit <JIP>. In GEN binary, <JIP> is at location <src1> and must be of type D (signed dword integer). <JIP> must be an immediate operand, it is a signed 32-bit number and is intended to be forward referencing. This value is added to IP pre-increment.</p> <p>If the <branch_ctrl> bit is set, then the <JIP> points to the first join instruction within the else block and <UIP> points to the endif instruction. If the <branch_ctrl> bit is not set, <JIP> and <UIP>, both point to endif.</p>			
<p>Format:</p> <p>else (<exec_size>) <JIP> <UIP> <branch_ctrl></p>			
Programming Notes			
<p>If all channels are redirected (by else or before else), relative jump is performed to the location specified by <JIP> + 1.</p>			
Restriction			
<p>Restriction : Predication is not allowed.</p>			
<p>Restriction : The execution size must be the same for the if, else, and endif instructions of the same code block.</p>			
<p>Restriction : The branch_ctrl must be set equal to (JIP != UIP).</p>			
Syntax			
<p>else (<exec_size>) imm32 imm32 branch_ctrl</p>			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < 32; n++) { if (WrEn.channel[n] == 1 branch_ctrl) { PcIP[n] = IP + JIP; } else { PcIP[n] = IP + UIP; } } if (PcIP != (IP+1)) { // for all channels Jump(IP + JIP); }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
N	N	N	N
DWord	Bit	Description	
0..3	127:96	JIP	



else - Else					
	<table border="1"> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td colspan="2">The byte-aligned jump distance if a jump is taken for the channel.</td> </tr> </table>	Format:	S31	The byte-aligned jump distance if a jump is taken for the channel.	
Format:	S31				
The byte-aligned jump distance if a jump is taken for the channel.					
95:64	<p>UIP</p> <table border="1"> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td colspan="2">The byte aligned jump distance if a jump is taken for the instruction.</td> </tr> </table>	Format:	S31	The byte aligned jump distance if a jump is taken for the instruction.	
Format:	S31				
The byte aligned jump distance if a jump is taken for the instruction.					
63:32	<p>Operand Control</p> <table border="1"> <tr> <td>Format:</td> <td>EU_INSTRUCTION_OPERAND_CONTROLS</td> </tr> </table>	Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
Format:	EU_INSTRUCTION_OPERAND_CONTROLS				
31:0	<p>Header</p> <table border="1"> <tr> <td>Format:</td> <td>EU_INSTRUCTION_HEADER</td> </tr> </table>	Format:	EU_INSTRUCTION_HEADER		
Format:	EU_INSTRUCTION_HEADER				



End If

endif - End If			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>The endif instruction terminates an if/else/endif block of code. It restores execution to the channels that were active prior to the if/else/endif block.</p> <p>The endif instruction is also used to hop out of nested conditionals by jumping to the end of the next outer conditional block when all channels are disabled.</p> <p>The following table describes the 32-bit JIP. In GEN binary, JIP is at location src1 and must be of type D (signed DWord integer). JIP must be an immediate operand, it is a signed 32-bit number. This value is added to IP pre-increment.</p>			
Format: endif JIP			
Restriction			
Restriction : Predication is not allowed.			
Restriction : The execution size must be the same for the if, else, and endif instructions of the same code block.			
Syntax			
endif (exec_size) imm32			
Pseudocode			
Evaluate(WrEn); if (WrEn == 0) { // all channels false Jump(IP + JIP); }			
Predication	Conditional Modifier	Saturation	Source Modifier
N	N	N	N
DWord	Bit	Description	
0.3	127:96	JIP Format: S31 Jump Target Offset. The relative offset in bytes if a jump is taken for the instruction.	
	95	Source 0 Address Immediate [9] Sign Bit	
	94:91	Src1.SrcType Format: SrcType	
	90:89	Src1.RegFile Format: RegFile	



endif - End If		
	88:64	Source 0
		Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode] == 'Align16')
		Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
	88:64	Source 0
	Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode] == 'Align1')	
	Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1	
63:32	Operand Control	
	Format: EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header	
	Format: EU_INSTRUCTION_HEADER	



Extended Math Function

math - Extended Math Function	
Source:	Eulsa
Length Bias:	4
<p>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.</p>	
Format:	[(pred)] math (exec_size) dst src0 src1 <FC>
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. INT DIV function does not support simultaneous write to two registers. INT DIV function does not support source modifiers.	
Restriction : The FDIV function is not supported in ALT_MODE.	
Restriction : The math instruction can use GRF or immediates as source. The source formats for immediates must be one of the source formats supported by math operation.	
Restriction : DepCtrl must not be used.	
Restriction : The math instruction must use GRF as destination.	
Restriction : The supported regioning mode for math instruction is align1 and align16. The following restrictions apply for align1 mode: Scalar source is supported. Source and destination horizontal stride must be the same. Regioning must ensure $Src.Vstride = Src.Width * Src.Hstride$. Source and destination offset must be the same, except the case of scalar source.	
Restriction : Half-float denorms are always retained.	
Restriction : Math Operation rules when float and half-floats are mixed between source or between source and destination operands. The half-float operand must be interleaved in the register for align1 and the source and destination register offset must be the same to DW granularity. For align16, the half-float operand is allowed to be packed.	
Restriction : The execution size must be no more than 8 when half-floats are used in source or destination operand.	
Restriction : The source operand must not span two registers if the destination operand spans just one register Example: Case (a) // Allowed. The source must be strided by 2. the offset is allowed to select between lower/upper 16b math (8) r10:f r11.0<16;8,2>:hf 0x01	



math - Extended Math Function

math (8) r10:f r11.1<16;8,2>:hf 0x01
 math (8) r10:f r11.0<16;8,2>:hf r12.1<16;8,2>:hf 0x09
 Case (b) // Allowed. The destination must be strided by 2. The offset is allowed to select between lower/upper 16b
 math (8) r10.0<2>:hf r11.0<8;8,1>:f
 math (8) r10.1<2>:hf r11.0<8;8,1>:f 0x01
 math (8) r10.0<2>:hf r11.0<16;8,2>:hf r12.0<16;8,2>:hf 0x09
 Case (c) // Allowed. Destination has stride of 2. The offset is allowed to select between upper/lower 16b
 math (8) r10.0<2>:hf r11.0<8;8,1>:f r12.1<16;8,2>:hf 0x09
 math (8) r10.1<2>:hf r11.1<16;8,2>:hf r12.0<8;8,1>:f 0x09
 Case (d) // Not Allowed. Destination is half-float but is not interleaved.
 math (8) r10.0<1>:hf r11.0<8;8,1>:f
 Case (e) // Not Allowed. Source is half-float but not interleaved
 math (8) r10.0<2>:hf r11.0<8;8,1>:f r12.0<8;8,1>:hf 0x09
 Case (f) // Not Allowed. Source operand spans 2 registers while destination spans one register.
 math (8) r83.8<1>:hf r12.4<4;4,1>:f null 0x02

Restriction : Math Operation rules when half-floats are used on both source and destination operands.

The execution size must be 8.

The half-float source must be packed or interleaved. When interleaving, both source and destination must be interleaved.

Example:

Case (a) // Allowed. The source and destination are packed or interleaved

math (8) r10.0:hf r11.0<8;8,1>:hf 0x01
 math (8) r10.0<2>:hf r11.0<16;8,2>:hf 0x01
 math (8) r10.8:hf r11.0<8;8,1>:hf 0x01
 math (8) r10.8<2>:hf r11.0<16;8,2>:hf 0x01

Restriction : For one source math operations src1 must be NULL.

Syntax

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

Pseudocode

```
Evaluate(WrEn);
for (n = 0; n < exec_size; n++) {
  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:
```



math - Extended Math Function

```

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
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	N	Y	Y

Src Types	Dst Types
F	F
D	D
UD	UD
F, HF	F, HF

DWord	Bit	Description
0..3	127:64	RegSource Format: EU_INSTRUCTION_SOURCES_REG_REG
	63:32	Operand Control Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:28	Controls B Format: EU_INSTRUCTION_CONTROLS_B
	27:24	Function Control (FC) Format: FC
	23:8	Controls A Format: EU_INSTRUCTION_CONTROLS_A
	7	Reserved Format: MBZ



math - Extended Math Function				
	6:0	Opcode		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">EU_OPCODE</td> </tr> </table>	Format:	EU_OPCODE
Format:	EU_OPCODE			



Find First Bit from LSB Side

fbl - Find First Bit from LSB Side			
Source:	Eulsa		
Length Bias:	4		
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.			
Restriction			
Restriction : No accumulator access, implicit or explicit.			
Syntax			
[(pred)] fbl (exec_size) reg reg [(pred)] fbl (exec_size) reg imm32			
Pseudocode			
<pre>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] = 0xFFFFFFFF; } else { dst.chan[n] = cnt; } } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types	Dst Types		
UD	UD		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	[(Operand Controls)][Src0.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG	
127:64	ImmSource		
	Exists If:	[(Operand Controls)][Src0.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	



fbl - Find First Bit from LSB Side					
31:0	<table border="1"> <tr> <td colspan="2">Header</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_HEADER</td> </tr> </table>	Header		Format:	EU_INSTRUCTION_HEADER
Header					
Format:	EU_INSTRUCTION_HEADER				



Find First Bit from MSB Side

fbh - Find First Bit from MSB Side			
Source:	Eulsa		
Length Bias:	4		
<p>If src0 is unsigned, the fbh instruction counts component-wise the leading zeros from src0 and stores the resulting counts in dst.</p> <p>If src0 is signed and positive, the fbh instruction counts component-wise the leading zeros from src0 and stores the resulting counts in dst.</p> <p>If src0 is signed and negative, the fbh instruction counts component-wise the leading ones from src0 and stores the resulting counts in dst.</p>			
Format:	[(pred)] fbh (exec_size) dst src0		
Programming Notes			
If src0 is zero, store 0xFFFFFFFF in dst.			
If src0 is signed and is -1 (0xFFFFFFFF), store 0xFFFFFFFF in dst.			
Restriction			
Restriction : No accumulator access, implicit or explicit.			
Syntax			
[(pred)] fbh (exec_size) reg reg [(pred)] fbh (exec_size) reg imm32			
Pseudocode			
<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] = 0xFFFFFFFF; } 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] == 0xFFFFFFFF) (src0.chan[n] == 0x00000000)) { dst.chan[n] = 0xFFFFFFFF; } else { dst.chan[n] = cnt; } } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types	Dst Types		
D, UD	UD		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([Operand Controls][Src0.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG



fbh - Find First Bit from MSB Side		
	127:64	ImmSource
		Exists If: ([Operand Controls][Src0.RegFile]== 'IMM')
	Format: EU_INSTRUCTION_SOURCES_IMM32	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Fraction

frc - Fraction			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p>			
<p>Format: [(pred)] frc[.cmod] (exec_size) dst src0</p>			
Syntax			
[(pred)] frc[.cmod] (exec_size) reg reg [(pred)] frc[.cmod] (exec_size) reg imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src0.chan[n] - floor(src0.chan[n]); } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	N	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([(Operand Controls][Src0.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource	
Exists If:		([(Operand Controls][Src0.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Goto

goto - Goto			
Source:	Eulsa		
Length Bias:	4		
<p>The goto instruction directs the instruction pointer to the offset specified by the UIP offset or to the next IP based on the BranchCtrl bit in the instruction. The active channels that are predicated on this instruction will take the IP + UIP path when BranchCtrl is set else the channels take IP + 1. The active channels that are not predicated on this instruction will be made inactive and waiting to be joined at the join IP. The join IP is IP + UIP when BranchCtrl is clear else it is the next IP.</p> <p>When there are no active channels the instruction pointer will move to IP + JIP.</p> <p>The goto instruction is used in conjunction with a join instruction. A goto deactivates some channels that are reactivated at some program-specified join instruction. See the join instruction for the activation rules.</p> <p>The goto and join instructions enable unstructured program control flow. These instructions must be used with additional care where dangling channels can result without proper compiler checks, meaning that it is expected that programs will navigate through these paths to reactivate the channels. Hardware does not provide native checks or reconvergence.</p> <p>The following table describes the two 32-bit instruction pointer offsets. Both the JIP and UIP are signed 32-bit numbers, added to IP pre-increment. In GEN binary, JIP and UIP are at locations src0 and src1 and must be of type DW (signed DWord integer).</p> <p>If SPF is ON, none of the PcIP are updated.</p>			
<p>Format:</p> <p>[(pred)] goto (exec_size) JIP UIP BranchCtrl</p>			
Restriction			
<p>Restriction : Cannot have a goto in the body and the corresponding join in the function or the subroutine. Similarly the brd and brc.</p>			
Syntax			
<p>[(pred)] goto (exec_size) imm32imm32 BranchCtrl</p>			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { // for the predicated active channels if (BranchCtrl) { PcIP[n] = IP + UIP; } else { PcIP[n] = IP + 1; } } else { // join IP, for the active non predicated channels if (BranchCtrl) { PcIP[n] = IP + 1; } else { PcIP[n] = IP + UIP; } } } if (BranchCtrl) { // if (PcIP != (IP + UIP)) { // for all channels if (PcIP != (IP + 1)) { // for all channels Jump(IP + JIP); } else { Jump(IP + 1); } } else { Jump(IP + UIP); } } else { // if (PcIP != (IP + 1)) { // for all channels Jump(IP + JIP); } else { Jump(IP + 1); } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N



goto - Goto				
DWord	Bit	Description		
0..3	127:96	JIP Format: <table border="1"><tr><td></td><td>S31</td></tr></table> The byte-aligned jump distance if a jump is taken for the channel.		S31
		S31		
	95:64	UIP Format: <table border="1"><tr><td></td><td>S31</td></tr></table> The byte aligned jump distance if a jump is taken for the instruction.		S31
		S31		
63:32	Operand Control Format: <table border="1"><tr><td></td><td>EU_INSTRUCTION_OPERAND_CONTROLS</td></tr></table>		EU_INSTRUCTION_OPERAND_CONTROLS	
	EU_INSTRUCTION_OPERAND_CONTROLS			
31:0	Header Format: <table border="1"><tr><td></td><td>EU_INSTRUCTION_HEADER</td></tr></table>		EU_INSTRUCTION_HEADER	
	EU_INSTRUCTION_HEADER			



GPGPU_CSR_BASE_ADDRESS

GPGPU_CSR_BASE_ADDRESS			
Source:	BSpec		
Length Bias:	2		
The GPGPU_CSR_BASE_ADDRESS command sets the base pointers for EU and L3 to Context Save and Restore EU State and SLM for GPGPU mid-thread preemption.			
Programming Notes			
Execution of this command causes a full pipeline flush, thus its use should be minimized for higher performance. State and instruction caches are flushed on completion of the flush.			
SW must always program PIPE_CONTROL with "CS Stall" and "Render Target Cache Flush Enable" set prior to programming GPGPU_CSR_BASE_ADDRESS command for GPGPU workloads i.e when pipeline select is GPGPU via PIPELINE_SELECT command. This is required to achieve better GPGPU preemption latencies for certain programming sequences. If programming PIPE_CONTROL has performance implications then preemption latencies can be trade off against performance by not implementing this programming note.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	Opcode
	28:27	Command SubType	
		Default Value:	0h GFXPIPE_COMMON
		Format:	Opcode
	26:24	3D Command Opcode	
		Default Value:	1h GFXPIPE_NONPIPELINED
		Format:	Opcode
	23:16	3D Command Sub Opcode	
		Default Value:	04h GPGPU_CSR_BASE_ADDRESS
		Format:	Opcode
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Format:	=n Total Length -2	
	Value	Name	Description
	1h	[Default]	Excludes DWord(0,1)



GPGPU_CSR_BASE_ADDRESS		
1..2	63:12	GPGPU CSR Base Address Format: GraphicsAddress[63:12] Specifies the 256K-byte aligned base address for GPGPU context GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].
	11:0	Reserved Format: MBZ



GPGPU_WALKER

GPGPU_WALKER			
Source:	RenderCS		
Length Bias:	2		
Programming Notes			
If the threads spawned by this command are required to observe memory writes performed by threads spawned from a previous command, software must precede this command with a command that performs a memory flush (e.g., MI_FLUSH).			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	1h GPGPU_WALKER
		Format:	OpCode
	23:16	SubOpcode	
Default Value:		05h GPGPU_WALKER SubOp	
Format:		OpCode	
15:11	Reserved		
	Format:	MBZ	
10	Indirect Parameter Enable		
	Format:	Enable	
If set, the values in DW 7, 10, 12 are ignored and replaced by the current values of the corresponding GPGPU_xxx MMIO registers:			
<ul style="list-style-type: none"> • GPGPU_DISPATCHDIMX (instead of DW7) • GPGPU_DISPATCHDIMY (instead of DW10) • GPGPU_DISPATCHDIMZ (instead of DW12) 			
9	Reserved		
	Format:	MBZ	



GPGPU_WALKER									
	8	<p>Predicate Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>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.</p>	Format:	Enable					
	Format:	Enable							
7:0	<p>DWord Length</p> <table border="1"> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0Dh</td> <td>DWORD_COUNT_n [Default]</td> <td>Excludes DWord (0,1)</td> </tr> </tbody> </table>	Format:	=n Total Length - 2	Value	Name	Description	0Dh	DWORD_COUNT_n [Default]	Excludes DWord (0,1)
Format:	=n Total Length - 2								
Value	Name	Description							
0Dh	DWORD_COUNT_n [Default]	Excludes DWord (0,1)							
1	31:8	Reserved							
	7:6	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
5:0	<p>Interface Descriptor Offset</p> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>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.</p>	Format:	U6						
Format:	U6								
2	31:17	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
16:0	<p>Indirect Data Length</p> <table border="1"> <tr> <td>Format:</td> <td>U17 in bytes</td> </tr> </table> <table border="1"> <thead> <tr> <th>Description</th> </tr> </thead> <tbody> <tr> <td>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. If Indirect Data is enabled, it is assumed that CURBE is not being used except for cross-thread constant data. This field must have the same alignment as the Indirect Object Data Start Address. It must be DQWord (32-byte) aligned.</td> </tr> <tr> <td>As the indirect data are sent directly to URB, the total size of Indirect data must be less than 63,488 (2048 URB lines - 64 lines for Interface Descriptors).</td> </tr> </tbody> </table>	Format:	U17 in bytes	Description	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. If Indirect Data is enabled, it is assumed that CURBE is not being used except for cross-thread constant data. 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, the total size of Indirect data must be less than 63,488 (2048 URB lines - 64 lines for Interface Descriptors).			
Format:	U17 in bytes								
Description									
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. If Indirect Data is enabled, it is assumed that CURBE is not being used except for cross-thread constant data. 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, the total size of Indirect data must be less than 63,488 (2048 URB lines - 64 lines for Interface Descriptors).									



GPGPU_WALKER														
3	31:6	<p>Indirect Data Start Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>IndirectObjectOffset[31:6]</td> </tr> </table> <p>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 on the mode of operation. It is the 64-byte aligned address of the indirect data.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0 - 512MB]</td> <td></td> <td>(Bits 31:29 MBZ)</td> </tr> </tbody> </table>	Format:	IndirectObjectOffset[31:6]	Value	Name	Description	[0 - 512MB]		(Bits 31:29 MBZ)				
	Format:	IndirectObjectOffset[31:6]												
Value	Name	Description												
[0 - 512MB]		(Bits 31:29 MBZ)												
5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ											
Format:	MBZ													
4	31:30	<p>SIMD Size</p> <p>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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SIMD8</td> <td>8 LSBs of the execution mask are used</td> </tr> <tr> <td>1</td> <td>SIMD16</td> <td>16 LSBs used in execution mask</td> </tr> <tr> <td>2</td> <td>SIMD32</td> <td>32 bits of execution mask used</td> </tr> </tbody> </table>	Value	Name	Description	0	SIMD8	8 LSBs of the execution mask are used	1	SIMD16	16 LSBs used in execution mask	2	SIMD32	32 bits of execution mask used
	Value	Name	Description											
	0	SIMD8	8 LSBs of the execution mask are used											
	1	SIMD16	16 LSBs used in execution mask											
	2	SIMD32	32 bits of execution mask used											
	29:22	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
	Format:	MBZ												
	21:16	<p>Thread Depth Counter Maximum</p> <p>The maximum value of the thread depth counter. Since the counter starts at 0, the depth is this value + 1. (Thread_Depth_Max+1)*(Thread_Height_Max+1)*(Thread_Width_Max+1) must equal Number of Threads in GPGPU Thread Group in the Interface Descriptor.</p>												
	15:14	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
	Format:	MBZ												
13:8	<p>Thread Height Counter Maximum</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U6-1</td> </tr> </table> <p>The maximum value of the thread height counter. The height is this value + 1.</p>	Format:	U6-1											
Format:	U6-1													
7:6	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ											
Format:	MBZ													
5:0	<p>Thread Width Counter Maximum</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U6-1</td> </tr> </table> <p>The maximum value of the thread width counter. The width is this value + 1.</p>	Format:	U6-1											
Format:	U6-1													
5	31:0	<p>Thread Group ID Starting X</p> <p>This is the initial value of the X component of the thread group. When X reaches the maximum value it rolls around to this value.</p>												



GPGPU_WALKER		
6	31:0	Reserved
		Format: MBZ
7	31:0	Thread Group ID X Dimension The X dimension of the thread group (maximum X is dimension -1)
8	31:0	Thread Group ID Starting Y This is the initial value of the Y component of the thread group. When Y reaches the maximum value it rolls around to this value.
9	31:0	Reserved
		Format: MBZ
10	31:0	Thread Group ID Y Dimension The Y dimension of the thread group (maximum Y is dimension -1)
11	31:0	Thread Group ID Starting/Resume Z This is the initial value of the Z component of the thread group.
12	31:0	Thread Group ID Z Dimension The Z dimension of the thread group (maximum Z is dimension -1)
13	31:0	Right Execution Mask The execution mask used for threads on the right (maximum X) of the thread group.
14	31:0	Bottom Execution Mask The execution mask used for threads on the bottom (maximum Y) of the thread group.



Half Precision HI8DS Render Target Write MSD

MSD_RTWH_HI8DS - Half Precision HI8DS Render Target Write MSD		
DWord	Bit	Description
Source: Render Cache DataPort		
Length Bias: 1		
Family: Other		
Group: Render Target R/W		
0	31	Reserved Format: MBZ Ignored
	30	Message Precision Subtype Default Value: 1h Format: Opcode Half precision data message
	29	Reserved Format: MBZ Ignored
	28:25	Message Length Format: U4 Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.
	24:20	Response Length Format: U5 Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.
	19	Header Present Format: MDC_MHP If set, indicates that the message includes the 2-register header.
	18	Reserved Format: MBZ Ignored



MSD_RTWH_HI8DS - Half Precision HI8DS Render Target Write MSD

17:14	Message Type	
	Default Value:	0Ch
	Format:	Opcode
Render Target Write message		
13	Per-Sample PS Enable	
	Format:	Enable
	<p>If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.</p> <p style="text-align: center;">Programming Notes</p> <p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE.</p> <p>When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_C0 as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0.</p> <p>When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples).</p> <p>When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>	
Last Render Target Select		
12	Format:	Enable
	<p>This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.</p> <p style="text-align: center;">Programming Notes</p> <p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set. In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>	
	Slot Group Select	
11	Format:	MDC_RT_SGS
	This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.	



MSD_RTWH_HI8DS - Half Precision HI8DS Render Target Write MSD

10:8	Render Target Message Subtype	
	Default Value:	3h
	Format:	Opcode
	SIMD8 dual source message. Use slots [15:8] for pixel enables, X/Y addresses, and oMask.	
	Programming Notes	
The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [31:24] are referenced instead of [15:8].		
7:0	Binding Table Index	
	Format:	MDC_BTS
Specifies the Binding Table Index for the message		



Half Precision LO8DS Render Target Write MSD

MSD_RTWH_LO8DS - Half Precision LO8DS Render Target Write MSD						
Source:	Render Cache DataPort					
Length Bias:	1					
Family:	Other					
Group:	Render Target R/W					
DWord	Bit	Description				
0	31	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ		
	Format:	MBZ				
	30	Message Precision Subtype <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Half precision data message	Default Value:	1h	Format:	Opcode
	Default Value:	1h				
	Format:	Opcode				
	29	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ		
	Format:	MBZ				
	28:25	Message Length <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.	Format:	U4		
Format:	U4					
24:20	Response Length <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.	Format:	U5			
Format:	U5					
19	Header Present <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> If set, indicates that the message includes the 2-register header.	Format:	MDC_MHP			
Format:	MDC_MHP					
18	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					



MSD_RTWH_LO8DS - Half Precision LO8DS Render Target Write MSD

17:14	Message Type	
	Default Value:	0Ch
	Format:	Opcode
Render Target Write message		
13	Per-Sample PS Enable	
	Format:	Enable
<p>If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.</p>		
Programming Notes		
<p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE.</p> <p>When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_C0 as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0.</p> <p>When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples).</p> <p>When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>		
12	Last Render Target Select	
	Format:	Enable
<p>This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.</p>		
Programming Notes		
<p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set. In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>		
11	Slot Group Select	
	Format:	MDC_RT_SGS
This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.		



MSD_RTWH_LO8DS - Half Precision LO8DS Render Target Write MSD

	10:8	Render Target Message Subtype	
		Default Value:	2h
		Format:	Opcode
		SIMD8 dual source message. Use slots [7:0] for pixel enables, X/Y addresses, and oMask.	
		Programming Notes	
The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [23:16] are referenced instead of [7:0].			
	7:0	Binding Table Index	
		Format:	MDC_BTS
		Specifies the Binding Table Index for the message	



Half Precision REP16 Render Target Write MSD

MSD_RTWH_REP16 - Half Precision REP16 Render Target Write MSD		
Source:	Render Cache DataPort	
Length Bias:	1	
Family:	Other	
Group:	Render Target R/W	
DWord	Bit	Description
0	31	Reserved Format: MBZ Ignored
	30	Message Precision Subtype Default Value: 1h Format: Opcode Half precision data message
	29	Reserved Format: MBZ Ignored
	28:25	Message Length Format: U4 Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.
	24:20	Response Length Format: U5 Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.
	19	Header Present Format: MDC_MHP If set, indicates that the message includes the 2-register header.
	18	Reserved Format: MBZ Ignored
	17:14	Message Type Default Value: 0Ch Format: Opcode Render Target Write message



MSD_RTWH_REP16 - Half Precision REP16 Render Target Write MSD

13	<p>Per-Sample PS Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.</p> <p style="text-align: center;">Programming Notes</p> <p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE.</p> <p>When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_C0 as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0.</p> <p>When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples).</p> <p>When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>	Format:	Enable
Format:	Enable		
12	<p>Last Render Target Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.</p> <p style="text-align: center;">Programming Notes</p> <p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set.</p> <p>In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>	Format:	Enable
Format:	Enable		
11	<p>Slot Group Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MDC_RT_SGS</td> </tr> </table> <p>This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.</p>	Format:	MDC_RT_SGS
Format:	MDC_RT_SGS		



MSD_RTWH_REP16 - Half Precision REP16 Render Target Write MSD

10:8	Render Target Message Subtype	
	Default Value:	1h
	Format:	Opcode
	SIMD16 Single source message with replicated data. Use slots [15:0] for pixel enables, X/Y addresses, and oMask.	
	Programming Notes	
	The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [31:16] are referenced instead of [15:0].	
7:0	Binding Table Index	
	Format:	MDC_BTS
	Specifies the Binding Table Index for the message	



Half Precision SIMD8 Render Target Write MSD

MSD_RTWH_SIMD8 - Half Precision SIMD8 Render Target Write MSD		
Source:	Render Cache DataPort	
Length Bias:	1	
Family:	Other	
Group:	Render Target R/W	
DWord	Bit	Description
0	31	Reserved Format: MBZ Ignored
	30	Message Precision Subtype Default Value: 1h Format: Opcode Half precision data message
	29	Reserved Format: MBZ Ignored
	28:25	Message Length Format: U4 Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.
	24:20	Response Length Format: U5 Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.
	19	Header Present Format: MDC_MHP If set, indicates that the message includes the 2-register header.
	18	Reserved Format: MBZ Ignored



MSD_RTWH_SIMD8 - Half Precision SIMD8 Render Target Write MSD

17:14	Message Type		
	Default Value:	0Ch	
	Format:	Opcode	
Render Target Write message			
13	Per-Sample PS Enable		
	Format:	Enable	
<p>If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.</p>			
Programming Notes			
<p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE.</p> <p>When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_C0 as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0.</p> <p>When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples).</p> <p>When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>			
12	Last Render Target Select		
	Format:	Enable	
<p>This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.</p>			
Programming Notes			
<p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set. In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>			
11	Slot Group Select		
	Format:	MDC_RT_SGS	
<p>This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.</p>			



MSD_RTWH_SIMD8 - Half Precision SIMD8 Render Target Write MSD

10:8	Render Target Message Subtype	
	Default Value:	4h
	Format:	Opcode
	SIMD8 single source message. Use slots [7:0] for pixel enables, X/Y addresses, and oMask.	
	Programming Notes	
The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [23:16] are referenced instead of [7:0].		
7:0	Binding Table Index	
	Format:	MDC_BTS
Specifies the Binding Table Index for the message		



Half Precision SIMD16 Render Target Write MSD

MSD_RTWH_SIMD16 - Half Precision SIMD16 Render Target Write MSD						
Source:	Render Cache DataPort					
Length Bias:	1					
Family:	Other					
Group:	Render Target R/W					
DWord	Bit	Description				
0	31	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ		
	Format:	MBZ				
	30	Message Precision Subtype <table border="1"> <tr> <td>Default Value:</td> <td>1h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> Half precision data message	Default Value:	1h	Format:	Opcode
	Default Value:	1h				
	Format:	Opcode				
	29	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ		
	Format:	MBZ				
	28:25	Message Length <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.	Format:	U4		
Format:	U4					
24:20	Response Length <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.	Format:	U5			
Format:	U5					
19	Header Present <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> If set, indicates that the message includes the 2-register header.	Format:	MDC_MHP			
Format:	MDC_MHP					
18	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					



MSD_RTWH_SIMD16 - Half Precision SIMD16 Render Target Write MSD

17:14	Message Type		
	Default Value:	0Ch	
	Format:	Opcode	
	Render Target Write message		
13	Per-Sample PS Enable		
	Format:	Enable	
	If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.		
	Programming Notes		
	<p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE.</p> <p>When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_C0 as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0.</p> <p>When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples).</p> <p>When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>		
12	Last Render Target Select		
	Format:	Enable	
	<p>This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.</p>		
	Programming Notes		
	<p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set. In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>		
11	Slot Group Select		
	Format:	MDC_RT_SGS	
	This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.		



MSD_RTWH_SIMD16 - Half Precision SIMD16 Render Target Write MSD

	10:8	Render Target Message Subtype	
		Default Value:	0h
		Format:	Opcode
		SIMD16 Single source message. Use slots [15:0] for pixel enables, X/Y addresses, and oMask.	
		Programming Notes	
		The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [31:16] are referenced instead of [15:0].	
	7:0	Binding Table Index	
		Format:	MDC_BTS
		Specifies the Binding Table Index for the message	



Halt

halt - Halt			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>The halt instruction temporarily suspends execution for all enabled compute channels. Upon execution, the enabled channels are sent to the instruction at (IP + UIP), if all channels are enabled at HALT, jump to the instruction at (IP + JIP).</p> <p>If the halt instruction is not inside any conditional code block, the values of JIP and UIP should be the same. If the halt instruction is inside a conditional code block, the UIP should be the end of the program and the JIP should be the end of the inner most conditional code block.</p> <p>The UIP must point to a HALT Instruction.</p> <p>If SPF is ON, the UIP must be used to update IP; JIP is not used in this case.</p> <p>The following table describes the two 32-bit instruction pointer offsets. Both the JIP and UIP are signed 32-bit numbers, added to IP pre-increment. In GEN binary, JIP and UIP are at locations src0 and src1 and must be of type DW (signed DWord integer). When the offsets are immediate, src0 regfile must be immediate and dst must be null.</p>			
<p>Format: [(pred)] halt (exec_size) JIP UIP</p>			
Syntax			
<pre>[(pred)] halt (exec_size) imm32 imm32</pre>			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < 32; n++) { if (WrEn.channel[n]) { PcIP[n] = IP + UIP; else { PcIP[n] = IP + 1; } } if (PcIP != (IP + 1)) { // for all channels Jump(IP + JIP); }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
DWord	Bit	Description	
0..3	127:96	JIP	
		Format:	S31
	The byte-aligned jump distance if a jump is taken for the channel.		
95:64	UIP		
	Format:	S31	
The byte aligned jump distance if a jump is taken for the instruction.			
63:32	Operand Control		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	



halt - Halt		
	31:0	Header
	Format:	EU_INSTRUCTION_HEADER



HCP_BSD_OBJECT

HCP_BSD_OBJECT			
Source:	VideoCS		
Length Bias:	2		
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HCP_BSD_OBJECT command fetches the HEVC bit stream for a slice starting with the first byte in the slice. The bit stream ends with the last non-zero bit of the frame and does not include any zero-padding at the end of the bit stream. There can be multiple slices in a HEVC frame and thus this command can be issued multiple times per frame.</p> <p>The HCP_BSD_OBJECT command must be the last command issued in the sequence of batch commands before the HCP starts decoding. Prior to issuing this command, it is assumed that all configuration parameters in the HCP have been loaded including workload configuration registers and configuration tables. When this command is issued, the HCP is waiting for bit stream data to be presented to the shift register.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
		Default Value:	7h Codec/Engine Name
		Format:	OpCode
		Codec/Engine Name = HCP = 7h	
	22:16	Media Instruction Command	
		Default Value:	20h HCP_BSD_OBJECT_STATE
		Format:	OpCode
	15:12	Reserved	
		Format:	MBZ
	11:0	Dword Length	
		Format:	=n
		(Excludes Dwords 0, 1).	
		Value	Name
	1h		



HCP_BSD_OBJECT								
1	31:0	Indirect BSD Data Length Format: U32						
		Specifies the length in bytes of the bitstream data for the current slice. It includes the first byte of the slice and the last non-zero byte of the in the slice. Specifically, the zero-padding bytes (if present) and the next start-code are excluded.						
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,268435455]</td> <td style="text-align: center;">Data_Length_with_28_bits_only</td> <td>Valid range is only from 0 to 268435455, which is corresponding to lower 28 bits.</td> </tr> </tbody> </table>	Value	Name	Description	[0,268435455]	Data_Length_with_28_bits_only	Valid range is only from 0 to 268435455, which is corresponding to lower 28 bits.
		Value	Name	Description				
[0,268435455]	Data_Length_with_28_bits_only	Valid range is only from 0 to 268435455, which is corresponding to lower 28 bits.						
2	31:29	Reserved Format: MBZ						
	28:0	Indirect Data Start Address Format: U29 Specifies the byte-aligned graphics memory starting address of the slice bit stream relative to the BSD Indirect Object Base Address .						



HCP_FQM_STATE

HCP_FQM_STATE																													
Source:	VideoCS																												
Length Bias:	2																												
<p>The HCP_FQM_STATE command loads the custom HEVC quantization tables into local RAM and may be issued up to 8 times: 4 scaling list per intra and inter.</p>																													
<p>Driver is responsible for performing the Scaling List division. So, save the division HW cost in HW. The 1/x value is provided in 16-bit fixed-point precision as $((1 \ll 17) / QM + 1) \gg 1$.</p>																													
<p>Note: FQM is computed as $(2^{16}) / QM$. If $QM=1$, FQM=all 1's.</p>																													
<p>To simplify the design, only a limited number of scaling lists are provided at the PAK interface: default two SizeID0 and two SizeID123 (one set for inter and the other set for intra), and the encoder only allows custom entries for these four matrices. The DC value of SizeID2 and SizeID3 will be provided.</p>																													
<p>When the scaling_list_enable_flag is set to disable, the scaling matrix is still sent to the PAK, and with all entries programmed to the same value of 16.</p>																													
<p>This is a picture level state command and is issued in encoding processes only.</p>																													
<p>DWords 2-33 form a table for the DCT coefficients, 2 16-bit coefficients/DWord.</p> <ul style="list-style-type: none"> • Size 4x4 for SizeID0, DWords 2-9. • Size 8x8 for SizeID1/2/3, DWords 2-33. 																													
<p>SizeID 0 (Table 4-13)</p> <table border="1"> <thead> <tr> <th>4x4</th> <th>[31:16]</th> <th>[15:0]</th> </tr> </thead> <tbody> <tr> <td>DWord 2</td> <td>AC(0,1)</td> <td>DC</td> </tr> <tr> <td>DWord 3</td> <td>AC(0,3)</td> <td>AC(0,2)</td> </tr> <tr> <td>DWord 4</td> <td>AC(1,1)</td> <td>AC(1,0)</td> </tr> <tr> <td>DWord 5</td> <td>AC(1,3)</td> <td>AC(1,2)</td> </tr> <tr> <td>DWord 6</td> <td>AC(2,1)</td> <td>AC(2,0)</td> </tr> <tr> <td>DWord 7</td> <td>AC(2,3)</td> <td>AC(2,2)</td> </tr> <tr> <td>DWord 8</td> <td>AC(3,1)</td> <td>AC(3,0)</td> </tr> <tr> <td>DWord 9</td> <td>AC(3,3)</td> <td>AC(3,2)</td> </tr> </tbody> </table>			4x4	[31:16]	[15:0]	DWord 2	AC(0,1)	DC	DWord 3	AC(0,3)	AC(0,2)	DWord 4	AC(1,1)	AC(1,0)	DWord 5	AC(1,3)	AC(1,2)	DWord 6	AC(2,1)	AC(2,0)	DWord 7	AC(2,3)	AC(2,2)	DWord 8	AC(3,1)	AC(3,0)	DWord 9	AC(3,3)	AC(3,2)
4x4	[31:16]	[15:0]																											
DWord 2	AC(0,1)	DC																											
DWord 3	AC(0,3)	AC(0,2)																											
DWord 4	AC(1,1)	AC(1,0)																											
DWord 5	AC(1,3)	AC(1,2)																											
DWord 6	AC(2,1)	AC(2,0)																											
DWord 7	AC(2,3)	AC(2,2)																											
DWord 8	AC(3,1)	AC(3,0)																											
DWord 9	AC(3,3)	AC(3,2)																											
<p>SizeID 1, 2, 3 (Table 4-14)</p> <table border="1"> <thead> <tr> <th>8x8</th> <th>[31:16]</th> <th>[15:0]</th> </tr> </thead> <tbody> <tr> <td>DWord 2</td> <td>AC(0,1)</td> <td>DC</td> </tr> <tr> <td>DWord 3</td> <td>AC(0,3)</td> <td>AC(0,2)</td> </tr> <tr> <td>DWord 4</td> <td>AC(0,5)</td> <td>AC(0,4)</td> </tr> <tr> <td>DWord 5</td> <td>AC(0,7)</td> <td>AC(0,6)</td> </tr> <tr> <td>DWord 6</td> <td>AC(1,1)</td> <td>AC(1,0)</td> </tr> </tbody> </table>			8x8	[31:16]	[15:0]	DWord 2	AC(0,1)	DC	DWord 3	AC(0,3)	AC(0,2)	DWord 4	AC(0,5)	AC(0,4)	DWord 5	AC(0,7)	AC(0,6)	DWord 6	AC(1,1)	AC(1,0)									
8x8	[31:16]	[15:0]																											
DWord 2	AC(0,1)	DC																											
DWord 3	AC(0,3)	AC(0,2)																											
DWord 4	AC(0,5)	AC(0,4)																											
DWord 5	AC(0,7)	AC(0,6)																											
DWord 6	AC(1,1)	AC(1,0)																											



HCP_FQM_STATE

DWord 7	AC(1,3)	AC(1,2)
DWord 8	AC(1,5)	AC(1,4)
DWord 9	AC(1,7)	AC(1,6)
...		
DWord 30	AC(7,1)	AC(7,0)
DWord 31	AC(7,3)	AC(7,2)
DWord 32	AC(7,5)	AC(7,4)
DWord 33	AC(7,7)	AC(7,6)

DWord	Bit	Description					
0	31:29	Command Type <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%;">Default Value:</td> <td>3h PARALLEL_VIDEO_PIPE</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	3h PARALLEL_VIDEO_PIPE	Format:	OpCode	
		Default Value:	3h PARALLEL_VIDEO_PIPE				
		Format:	OpCode				
	28:27	Pipeline Type <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Default Value:</td> <td>2h</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	2h	Format:	OpCode	
		Default Value:	2h				
	Format:	OpCode					
	26:23	Media Instruction Opcode <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%;">Default Value:</td> <td>7h Codec/Engine Name</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table> Codec/Engine Name = HCP = 7h	Default Value:	7h Codec/Engine Name	Format:	OpCode	
Default Value:		7h Codec/Engine Name					
Format:		OpCode					
22:16	Media Instruction Command <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%;">Default Value:</td> <td>5h HCP_FQM_STATE</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	5h HCP_FQM_STATE	Format:	OpCode		
	Default Value:	5h HCP_FQM_STATE					
Format:	OpCode						
15:12	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ					
11:0	Dword Length <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>=n</td> </tr> </table> (Excludes Dwords 0, 1). <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">20h</td> <td></td> </tr> </tbody> </table>	Format:	=n	Value	Name	20h	
Format:	=n						
Value	Name						
20h							
1	31:16	FQM DC Value: (1/DC): <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U16</td> </tr> </table>	Format:	U16			
		Format:	U16				
		Specifies DC value of the scaling list for 16x16 (SizeID=2) or 32x32 (SizeID=3).					
		DC Value = scaling_list_dc_coef_minus8 + 8.					
Driver will do the division.							



HCP_FQM_STATE												
	15:5	Reserved Format: U11										
	4:3	Color Component Format: U2 Luma and Chroma's share the same scaling list and DC value for the same SizeID. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Luma</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Chroma Cb</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Chroma Cr</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	0	Luma	1	Chroma Cb	2	Chroma Cr	3	Reserved
	Value	Name										
	0	Luma										
	1	Chroma Cb										
	2	Chroma Cr										
	3	Reserved										
	2:1	SizeID Format: U2 <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>SizeID 0 4x4</td> </tr> <tr> <td style="text-align: center;">1</td> <td>SizeID 1, 2, 3 (8x8, 16x16, 32x32)</td> </tr> <tr> <td style="text-align: center;">2</td> <td>SizeID 2 (for DC value in 16x16)</td> </tr> <tr> <td style="text-align: center;">3</td> <td>SizeID 3 (for DC value in 32x32)</td> </tr> </tbody> </table>	Value	Name	0	SizeID 0 4x4	1	SizeID 1, 2, 3 (8x8, 16x16, 32x32)	2	SizeID 2 (for DC value in 16x16)	3	SizeID 3 (for DC value in 32x32)
	Value	Name										
	0	SizeID 0 4x4										
1	SizeID 1, 2, 3 (8x8, 16x16, 32x32)											
2	SizeID 2 (for DC value in 16x16)											
3	SizeID 3 (for DC value in 32x32)											
0	Intra/Inter Format: U1 This field specifies the quant matrix intra or inter type. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Intra</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Inter</td> </tr> </tbody> </table>	Value	Name	0	Intra	1	Inter					
Value	Name											
0	Intra											
1	Inter											
2..33	1023:0	See Tables 4-13 and 4-14										



HCP_IND_OBJ_BASE_ADDR_STATE

HCP_IND_OBJ_BASE_ADDR_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HCP_IND_OBJ_BASE_ADDR_STATE command is used to define the indirect object base address of the stream in graphics memory. This is a frame level command. (Is it frame or picture level?) This is a picture level state command and is issued in both encoding and decoding processes.</p>			
Compressed Header Format			
Fields	Bits		
Bin	0	Kernel Binarized Syntax	
Probability select	1	0 -> indicates probability 128 1 -> indicates probability 256	
	Repeat to pack a Cacheline		
Partition1 and TileSize record			
Fields	Bits		
Tile Size	31:0	Partition1 Size is 16-bit value, Tile Size is 32-bit value	
AddressOffset	63:32	Cacheline Address Offset to be Modified	
Offset	69:64	Byte offset to be Modified	
16-bit vs 32-bit update	70	0: Update 16-bit; 1: Update 32-bit	
Reserved	511:71		
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
		Default Value:	7h Codec/Engine Name
		Format:	OpCode
		Codec/Engine Name = HCP = 7h	
	22:16	Media Instruction Command	
		Default Value:	3h HCP_IND_OBJ_BASE_ADDR_STATE
		Format:	OpCode



HCP_IND_OBJ_BASE_ADDR_STATE											
	<table border="1"> <tr> <td style="text-align: center;">15:12</td> <td>Reserved</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	15:12	Reserved	Format:	MBZ						
15:12	Reserved										
Format:	MBZ										
	<table border="1"> <tr> <td style="text-align: center;">11:0</td> <td>Dword Length</td> </tr> <tr> <td>Format:</td> <td>=n</td> </tr> <tr> <td colspan="2">(Excludes Dwords 0, 1).</td> </tr> <tr> <td style="text-align: center;">Value</td> <td style="text-align: center;">Name</td> </tr> <tr> <td>Ch</td> <td></td> </tr> </table>	11:0	Dword Length	Format:	=n	(Excludes Dwords 0, 1).		Value	Name	Ch	
11:0	Dword Length										
Format:	=n										
(Excludes Dwords 0, 1).											
Value	Name										
Ch											
1..2	<table border="1"> <tr> <td style="text-align: center;">63:0</td> <td>HCP Indirect Bitstream Object Base Address</td> </tr> <tr> <td>Format:</td> <td>SplitBaseAddress4KByteAligned</td> </tr> <tr> <td colspan="2">Specifies the 4K-byte aligned memory base address for the read-only indirect data object pointed in the HCP_BSD_OBJECT command for fetching (reading) the compressed Slice Data.</td> </tr> </table>	63:0	HCP Indirect Bitstream Object Base Address	Format:	SplitBaseAddress4KByteAligned	Specifies the 4K-byte aligned memory base address for the read-only indirect data object pointed in the HCP_BSD_OBJECT command for fetching (reading) the compressed Slice Data.					
63:0	HCP Indirect Bitstream Object Base Address										
Format:	SplitBaseAddress4KByteAligned										
Specifies the 4K-byte aligned memory base address for the read-only indirect data object pointed in the HCP_BSD_OBJECT command for fetching (reading) the compressed Slice Data.											
3	<table border="1"> <tr> <td style="text-align: center;">31:0</td> <td>HCP Indirect Bitstream Object Memory Address Attributes</td> </tr> <tr> <td>Format:</td> <td>MemoryAddressAttributes</td> </tr> </table>	31:0	HCP Indirect Bitstream Object Memory Address Attributes	Format:	MemoryAddressAttributes						
31:0	HCP Indirect Bitstream Object Memory Address Attributes										
Format:	MemoryAddressAttributes										
4..5	<table border="1"> <tr> <td style="text-align: center;">63:0</td> <td>HCP Indirect Bitstream Object Access Upper Bound</td> </tr> <tr> <td>Format:</td> <td>SplitBaseAddress4KByteAligned</td> </tr> <tr> <td colspan="2">Decoder only.</td> </tr> <tr> <td colspan="2">This field specifies the 4K-byte aligned maximum memory address access by the indirect data object in the HCP_BSD_OBJECT command for the slice bit stream. Indirect data accessed at this address or greater will cause the HCP to stop issuing requests to the GAC and the BSP VLD will then only receive zeros until a slice done is received.</td> </tr> <tr> <td colspan="2">Setting this field to 0 will cause this range to be ignored by the HCP.</td> </tr> </table>	63:0	HCP Indirect Bitstream Object Access Upper Bound	Format:	SplitBaseAddress4KByteAligned	Decoder only.		This field specifies the 4K-byte aligned maximum memory address access by the indirect data object in the HCP_BSD_OBJECT command for the slice bit stream. Indirect data accessed at this address or greater will cause the HCP to stop issuing requests to the GAC and the BSP VLD will then only receive zeros until a slice done is received.		Setting this field to 0 will cause this range to be ignored by the HCP.	
63:0	HCP Indirect Bitstream Object Access Upper Bound										
Format:	SplitBaseAddress4KByteAligned										
Decoder only.											
This field specifies the 4K-byte aligned maximum memory address access by the indirect data object in the HCP_BSD_OBJECT command for the slice bit stream. Indirect data accessed at this address or greater will cause the HCP to stop issuing requests to the GAC and the BSP VLD will then only receive zeros until a slice done is received.											
Setting this field to 0 will cause this range to be ignored by the HCP.											
6..7	<table border="1"> <tr> <td style="text-align: center;">63:0</td> <td>HCP Indirect CU Object Base Address</td> </tr> <tr> <td>Format:</td> <td>BaseAddress4KByteAligned</td> </tr> <tr> <td colspan="2">Encoder only.</td> </tr> <tr> <td colspan="2">Specifies the 4K-byte aligned data buffer base address for the read-only indirect data object for fetching (reading) per CU data during the encoding process.</td> </tr> </table>	63:0	HCP Indirect CU Object Base Address	Format:	BaseAddress4KByteAligned	Encoder only.		Specifies the 4K-byte aligned data buffer base address for the read-only indirect data object for fetching (reading) per CU data during the encoding process.			
63:0	HCP Indirect CU Object Base Address										
Format:	BaseAddress4KByteAligned										
Encoder only.											
Specifies the 4K-byte aligned data buffer base address for the read-only indirect data object for fetching (reading) per CU data during the encoding process.											
8	<table border="1"> <tr> <td style="text-align: center;">31:0</td> <td>HCP Indirect CU Object Object Memory Address Attributes</td> </tr> <tr> <td>Format:</td> <td>MemoryAddressAttributes</td> </tr> </table>	31:0	HCP Indirect CU Object Object Memory Address Attributes	Format:	MemoryAddressAttributes						
31:0	HCP Indirect CU Object Object Memory Address Attributes										
Format:	MemoryAddressAttributes										
9..10	<table border="1"> <tr> <td style="text-align: center;">63:0</td> <td>HCP PAK-BSE Object Base Address</td> </tr> <tr> <td>Format:</td> <td>BaseAddress4KByteAligned</td> </tr> <tr> <td colspan="2">Encoder only.</td> </tr> <tr> <td colspan="2">Specifies the 4K-byte aligned memory base address for the write-only data pointed by the PAK engine for writing out the compressed bitstream.</td> </tr> </table>	63:0	HCP PAK-BSE Object Base Address	Format:	BaseAddress4KByteAligned	Encoder only.		Specifies the 4K-byte aligned memory base address for the write-only data pointed by the PAK engine for writing out the compressed bitstream.			
63:0	HCP PAK-BSE Object Base Address										
Format:	BaseAddress4KByteAligned										
Encoder only.											
Specifies the 4K-byte aligned memory base address for the write-only data pointed by the PAK engine for writing out the compressed bitstream.											



HCP_IND_OBJ_BASE_ADDR_STATE				
11	31:0	<p>HCP PAK-BSE Object Address Memory Address Attributes</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>MemoryAddressAttributes</td> </tr> </table> <p>Encoder only.</p>	Format:	MemoryAddressAttributes
Format:	MemoryAddressAttributes			
12..13	63:0	<p>HCP PAK-BSE Object Access Upper Bound</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>SplitBaseAddress4KByteAligned</td> </tr> </table> <p>Encoder only.</p> <p>This field specifies the 4K-byte aligned maximum memory address access by the HCP_PAK_OBJECT command for writing out the slice bit stream. Indirect data accessed at this address and beyond will be blocked by the hardware and ignored.</p> <p>This address must be greater than the HCP PAK-BSE Object Base Address state.</p>	Format:	SplitBaseAddress4KByteAligned
Format:	SplitBaseAddress4KByteAligned			



HCP_PAK_INSERT_OBJECT

HCP_PAK_INSERT_OBJECT	
Source:	VideoCS
Length Bias:	2
<p>It is an encoder only command, operating at bitstream level, before and after SliceData compressed bitstream. It is setup by the header and tail present flags in the Slice State command. If these flags are set and no subsequent PAK_INSERT_OBJECT commands are issued, the pipeline will hang.</p>	
<p>The HCP_PAK_INSERT_OBJECT command supports both inline and indirect data payload, but only one can be active at any time. It is issued to insert a chunk of bits (payload) into the current compressed bitstream output buffer (specified in the HCP_PAK-BSE Object Base Address field of the HCP_IND_OBJ_BASE_ADDR_STATE command) starting at its current write pointer bit position. Hardware will keep track of this write pointer's byte position and the associated next bit insertion position index.</p>	
<p>It is a variable length command when the payload (data to be inserted) is presented as inline data within the command itself. The inline payload is a multiple of 32-bit (1 DW), as the data bus to the compressed bitstream output buffer is 32-bit wide.</p>	
<p>The payload 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. Note that : Stitch Command is used if the beginning position of data is in bit position. When PAK Insert Command is used the beginning position must be in byte position.</p>	
<p>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 bitstream.</p>	
<p>Internally, HCP 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.</p>	
<p>The payload 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.</p>	
<p>Data to be inserted can be a valid NAL units or a partial NAL unit. It can be any encoded syntax elements bit data before the encoded Slice Data (PAK Object Command) of the current Slice - SPS NAL, PPS NAL, SEI NAL and Other Non-Slice NAL, Leading_Zero_8_bits (as many bytes as there is), Start Code , Slice Header. Any 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 bitstream, whichever comes first Cabac_Zero_Word or Trailing_Zero_8bits (as many bytes as there is).</p>	
<p>Certain NAL unit has a minimum byte size requirement. As such the hardware will optionally (enabled by SLICE 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.</p>	
<p>Context switch interrupt is not supported by this command.</p>	



DWord	Bit	Description								
0	31:29	Command Type								
		Default Value: 3h PARALLEL_VIDEO_PIPE Format: OpCode								
	28:27	Pipeline Type								
		Default Value: 2h Format: OpCode								
	26:23	Media Instruction Opcode								
		Default Value: 7h Codec/Engine Name Format: OpCode Codec/Engine Name = HCP = 7h								
22:16	Media Instruction Command									
	Default Value: 22h HCP_PAK_INSERT_OBJECT Format: OpCode									
15:12	Reserved									
	Format: MBZ									
11:0	Dword Length									
	Format: =n (Excludes Dwords 0, 1) = N-1, Variable Length in DW if inline data; =4, fixed length if indirect data.									
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[1,4095]</td> <td></td> </tr> </tbody> </table>	Value	Name	[1,4095]						
	Value	Name								
[1,4095]										
1	31	Indirect Payload Enable								
		Format: Enable								
		Only one of these two payload modes can be active at any time. When Slice Size Conformance is enable the Payload(header) must be inline only so this bit set to MBZ.								
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>inline payload is used</td> <td></td> </tr> <tr> <td>1</td> <td>indirect payload is used</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0	inline payload is used		1	indirect payload is used	
	Value	Name	Description							
0	inline payload is used									
1	indirect payload is used									
30:18	Reserved									
	Format: MBZ									
17:16	DataByteOffset - SrcDataStartingByteOffset[1:0]									
	Format: U2 Source Data Starting Byte Position within the very first inline DW.									



HCP_PAK_INSERT_OBJECT																				
15	HeaderLengthExcludeFrmSize																			
	Format:	U1																		
	<p>In case this flag is on, bits are NOT accumulated during current access unit coding neither for Cabac Zero Word insertion bits counting or for output in MMIO register HCP_BITSTREAM_BYTECOUNT_FRAME_NO_HEADER.</p> <p>When using HeaderLenghtExcludeFrmSize for header insertion, the software needs to make sure that data comes already with inserted start code emulation bytes. SW shouldn't set EmulationFlag bit (Bit 3 of DWORD1 of HCP_PAK_INSERT_OBJECT).</p>																			
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>All bits accumulated</td> </tr> <tr> <td>1</td> <td>Bits during current call are not accumulated</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>All bits accumulated</td> <td></td> </tr> <tr> <td>1</td> <td>Bits during current call are not accumulated</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: center;">Must be set to 0 for JPEG encoder.</td> </tr> </tbody> </table>		Value	Description	0	All bits accumulated	1	Bits during current call are not accumulated	Value	Name	Description	0	All bits accumulated		1	Bits during current call are not accumulated		Programming Notes		Must be set to 0 for JPEG encoder.
Value	Description																			
0	All bits accumulated																			
1	Bits during current call are not accumulated																			
Value	Name	Description																		
0	All bits accumulated																			
1	Bits during current call are not accumulated																			
Programming Notes																				
Must be set to 0 for JPEG encoder.																				
13:8	DataBitsInLastDW - SrCDataEndingBitInclusion[5:0]																			
	Format:	U6																		
<p>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.</p> <p>For VP9: The Driver has to give byte aligned Data for the last inline DW. (the driver pads Zeros to next byte boundary to the original header if it was not byte aligned on the last inline DW).</p>																				
7:4	SkipEmulByteCnt - Skip Emulation Byte Count																			
	Format:	U4																		
<p>Skip emulation check for number of starting bytes It can be programmed from 0 to 15 bytes. For example, to skip the start code that has already prefixed in the bitstream.</p> <p>Only valid for HEVC and reserved for VP9.</p>																				
3	EmulationFlag - EmulationByteBitsInsertEnable																			
	Format:	U1																		
	<p>Only valid for HEVC and reserved for VP9.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>StartCodePrefix</td> <td>Instruct the hardware to perform Start Code Prefix (0x 00 00 01/02/03/00) Search and Prevention Byte (0x 03) insertion on the</td> </tr> </tbody> </table>		Value	Name	Description	1	StartCodePrefix	Instruct the hardware to perform Start Code Prefix (0x 00 00 01/02/03/00) Search and Prevention Byte (0x 03) insertion on the												
Value	Name	Description																		
1	StartCodePrefix	Instruct the hardware to perform Start Code Prefix (0x 00 00 01/02/03/00) Search and Prevention Byte (0x 03) insertion on the																		



HCP_PAK_INSERT_OBJECT						
		insertion data of this command. It is required that hardware will handle a start code prefix crossing the boundary between.				
	2	InsertionCommand Insertion commands, or an insertion command followed by a PAK Object command.				
	2	<p>LastHeaderFlag - LastSrcHeaderDataInsertCommandFlag</p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>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.</p>	Format:	U1		
Format:	U1					
	1	<p>EndOfSliceFlag - LastDstDataInsertCommandFlag</p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>No more insertion command and no more PAK-OBJECT command follows. Flush data out to memory.</p>	Format:	U1		
Format:	U1					
	0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					
2..n	127:0	<p>Indirect Payload</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Indirect Payload Enable]= =1)</td> </tr> <tr> <td>Format:</td> <td>HCP_PAK_INSERT_OBJECT_INDIRECT_PAYLOAD</td> </tr> </table>	Exists If:	(([Indirect Payload Enable]= =1)	Format:	HCP_PAK_INSERT_OBJECT_INDIRECT_PAYLOAD
Exists If:	(([Indirect Payload Enable]= =1)					
Format:	HCP_PAK_INSERT_OBJECT_INDIRECT_PAYLOAD					
	31:0	<p>Inline PayLoad</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Indirect Payload Enable]= =0)</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Actual Data (inline) to be inserted to the output bitstream buffer.</p>	Exists If:	(([Indirect Payload Enable]= =0)	Format:	U32
Exists If:	(([Indirect Payload Enable]= =0)					
Format:	U32					



HCP_PAK_OBJECT

HCP_PAK_OBJECT				
Source:	VideoCS			
Length Bias:	2			
This is a per-LCU command for encoder only.				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	3h PARALLEL_VIDEO_PIPE	
		Format:	OpCode	
	28:27	28:27	Pipeline Type	
			Default Value:	2h
			Format:	OpCode
	26:23	26:23	Media Instruction Opcode	
			Default Value:	7h Codec/Engine Name
			Format:	OpCode
			Codec/Engine Name = HCP = 7h	
22:16	22:16	Media Instruction Command		
		Default Value:	21h HCP_PAK_OBJECT	
		Format:	OpCode	
15:12	15:12	Reserved		
		Format:	MBZ	
11:0	11:0	Dword Length		
		Format:	=n	
		(Excludes Dwords 0, 1).		
		Value	Name	
		1h		
3h				
1	31	LastCtbOfSlice Flag		
		Format:	Boolean	
		Value	Name	
		0	False	
		1	True	



HCP_PAK_OBJECT		
	30	LastCtbOfTile Flag Format: U1 Indicates last LCU or SB of a Tile
	29:24	CU_count_minus1 Format: U6 Number of CUs in the current LCU = CU_count_minus1 + 1. Minimum, there must be 1 CU in a LCU.
	23:21	Reserved Format: MBZ
	20:0	split_coding_unit_flag[x0][y0] Format: Split_coding_unit_flags If CU size=64x64 and DQP!=0, split should happen at least once
2	31:16	Current LCU Y Addr Format: U16
	15:0	Current LCU X Addr Format: U16
3..4	31:0	Reserved Format: MBZ



HCP_PIC_STATE

HCP_PIC_STATE				
Source:	VideoCS			
Length Bias:	2			
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>This is a picture level command and is issued only once per workload for both encoding and decoding processes.</p>				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	3h PARALLEL_VIDEO_PIPE	
		Format:	OpCode	
	28:27	28:27	Pipeline Type	
			Default Value:	2h
			Format:	OpCode
	26:23	26:23	Media Instruction Opcode	
			Default Value:	7h Codec/Engine Name
			Format:	OpCode
	Codec/Engine Name = HCP = 7h			
22:16	22:16	Media Instruction Command		
		Default Value:	10h HCP_PIC_STATE	
		Format:	OpCode	
15:12	15:12	Reserved		
		Format:	MBZ	
11:0	11:0	Dword Length		
		Format:	=n	
		(Excludes Dwords 0, 1).		
		Value	Name	
		11h		
1Dh				
1	31:26	Reserved		
		Format:	MBZ	
		25:16	25:16	FrameHeightInMinCbMinus1
Format:	U10			
Specifies the height of each decoded picture in units of minimum coding block size.				



HCP_PIC_STATE											
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0-4122]</td> <td></td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2"> The decoded picture height in units of luma samples equals <ul style="list-style-type: none"> • (FrameHeightInMinCbMinus1 + 1) * • (1 << (log2_min_coding_block_size_minus3 + 3)) </td> </tr> <tr> <td colspan="2"> In HEVC Encoder mode, the following restrictions apply. Note : for a frame height that is not an integer multiple of LCU : Kernel : on last LCU at frame's bottom edge, it must align to CU boundary. This applies to all size of LCU: 16x16, 32x32, 64x64. Kernel does not count/include CUs outside of the picture in PakObj/CU_record respectively. Driver : sets up a LCU aligned (both in X/Y direction) surface. </td> </tr> </tbody> </table>	Value	Name	[0-4122]		Programming Notes		The decoded picture height in units of luma samples equals <ul style="list-style-type: none"> • (FrameHeightInMinCbMinus1 + 1) * • (1 << (log2_min_coding_block_size_minus3 + 3)) 		In HEVC Encoder mode, the following restrictions apply. Note : for a frame height that is not an integer multiple of LCU : Kernel : on last LCU at frame's bottom edge, it must align to CU boundary. This applies to all size of LCU: 16x16, 32x32, 64x64. Kernel does not count/include CUs outside of the picture in PakObj/CU_record respectively. Driver : sets up a LCU aligned (both in X/Y direction) surface.	
Value	Name										
[0-4122]											
Programming Notes											
The decoded picture height in units of luma samples equals <ul style="list-style-type: none"> • (FrameHeightInMinCbMinus1 + 1) * • (1 << (log2_min_coding_block_size_minus3 + 3)) 											
In HEVC Encoder mode, the following restrictions apply. Note : for a frame height that is not an integer multiple of LCU : Kernel : on last LCU at frame's bottom edge, it must align to CU boundary. This applies to all size of LCU: 16x16, 32x32, 64x64. Kernel does not count/include CUs outside of the picture in PakObj/CU_record respectively. Driver : sets up a LCU aligned (both in X/Y direction) surface.											
15	<p>PAK Transform Skip Enable If global bit, transform_skip_enabled_flag in image state dw4.22 set to 1 and this bit set to, 1 -> HW will perform transform skip and over write the ENC decision in CU record 0-> HW will not perform transform skip if transform_skip_enabled_flag=0, no transform skip performed in HW or ENC</p> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">Encoder Only</td> </tr> </tbody> </table>	Programming Notes		Encoder Only							
Programming Notes											
Encoder Only											
14:10	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
9:0	<p>FrameWidthInMinCbMinus1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U10</td> </tr> </table> <p>Specifies the width of each decoded picture in units of minimum coding block size.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0-4122]</td> <td></td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2"> The decoded picture width in units of luma samples equals <ul style="list-style-type: none"> • (FrameWidthInMinCbMinus1 + 1) * • (1 << (log2_min_coding_block_size_minus3 + 3)) </td> </tr> </tbody> </table>	Format:	U10	Value	Name	[0-4122]		Programming Notes		The decoded picture width in units of luma samples equals <ul style="list-style-type: none"> • (FrameWidthInMinCbMinus1 + 1) * • (1 << (log2_min_coding_block_size_minus3 + 3)) 	
Format:	U10										
Value	Name										
[0-4122]											
Programming Notes											
The decoded picture width in units of luma samples equals <ul style="list-style-type: none"> • (FrameWidthInMinCbMinus1 + 1) * • (1 << (log2_min_coding_block_size_minus3 + 3)) 											
2	<p style="text-align: center;">15</p> <p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table> <p style="text-align: center;">31:12</p> <p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ	Format:	MBZ						
Format:	MBZ										
Format:	MBZ										



HCP_PIC_STATE												
	11:10	MaxPCMSize Format: U2 Specifies the largest allowed PCM coding block size. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">3</td> <td>Reserved</td> </tr> <tr> <td style="text-align: center;">2</td> <td>32x32</td> </tr> <tr> <td style="text-align: center;">1</td> <td>16x16</td> </tr> <tr> <td style="text-align: center;">0</td> <td>8x8</td> </tr> </tbody> </table>	Value	Name	3	Reserved	2	32x32	1	16x16	0	8x8
	Value	Name										
	3	Reserved										
	2	32x32										
	1	16x16										
	0	8x8										
	9:8	MinPCMSize Format: U2 Specifies the smallest allowed PCM coding block size. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">3</td> <td>Reserved</td> </tr> <tr> <td style="text-align: center;">2</td> <td>32x32</td> </tr> <tr> <td style="text-align: center;">1</td> <td>16x16</td> </tr> <tr> <td style="text-align: center;">0</td> <td>8x8</td> </tr> </tbody> </table>	Value	Name	3	Reserved	2	32x32	1	16x16	0	8x8
	Value	Name										
	3	Reserved										
	2	32x32										
	1	16x16										
	0	8x8										
	7:6	MaxTUSize Format: U2 Specifies the largest allowed transform block size. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">3</td> <td>32x32</td> </tr> <tr> <td style="text-align: center;">2</td> <td>16x16</td> </tr> <tr> <td style="text-align: center;">1</td> <td>8x8</td> </tr> <tr> <td style="text-align: center;">0</td> <td>4x4</td> </tr> </tbody> </table>	Value	Name	3	32x32	2	16x16	1	8x8	0	4x4
	Value	Name										
	3	32x32										
	2	16x16										
	1	8x8										
	0	4x4										
	5:4	MinTUSize Format: U2 Specifies the smallest allowed transform block size. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">3</td> <td>32x32</td> </tr> <tr> <td style="text-align: center;">2</td> <td>16x16</td> </tr> <tr> <td style="text-align: center;">1</td> <td>8x8</td> </tr> <tr> <td style="text-align: center;">0</td> <td>4x4</td> </tr> </tbody> </table>	Value	Name	3	32x32	2	16x16	1	8x8	0	4x4
	Value	Name										
3	32x32											
2	16x16											
1	8x8											
0	4x4											
3:2	CtbSize (LCUSize) Format: U2 Specifies the coding tree block size. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> </tbody> </table>	Value	Name									
Value	Name											



HCP_PIC_STATE																					
	<table border="1"> <tr> <td>3</td> <td>64x64</td> </tr> <tr> <td>2</td> <td>32x32</td> </tr> <tr> <td>1</td> <td>16x16</td> </tr> <tr> <td>0</td> <td>illegal/reserved</td> </tr> </table> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Programming Notes</td> </tr> <tr> <td>LCU is restricted based on the picture size.</td> </tr> </table>	3	64x64	2	32x32	1	16x16	0	illegal/reserved	Programming Notes	LCU is restricted based on the picture size.										
3	64x64																				
2	32x32																				
1	16x16																				
0	illegal/reserved																				
Programming Notes																					
LCU is restricted based on the picture size.																					
	<table border="1"> <tr> <td>1:0</td> <td>MinCUSize</td> </tr> <tr> <td></td> <td>Format: U2</td> </tr> <tr> <td></td> <td>Specifies the smallest coding block size.</td> </tr> <tr> <td></td> <td> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>64x64</td> </tr> <tr> <td>2</td> <td>32x32</td> </tr> <tr> <td>1</td> <td>16x16</td> </tr> <tr> <td>0</td> <td>8x8</td> </tr> </tbody> </table> </td> </tr> </table>	1:0	MinCUSize		Format: U2		Specifies the smallest coding block size.		<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>64x64</td> </tr> <tr> <td>2</td> <td>32x32</td> </tr> <tr> <td>1</td> <td>16x16</td> </tr> <tr> <td>0</td> <td>8x8</td> </tr> </tbody> </table>	Value	Name	3	64x64	2	32x32	1	16x16	0	8x8		
1:0	MinCUSize																				
	Format: U2																				
	Specifies the smallest coding block size.																				
	<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>64x64</td> </tr> <tr> <td>2</td> <td>32x32</td> </tr> <tr> <td>1</td> <td>16x16</td> </tr> <tr> <td>0</td> <td>8x8</td> </tr> </tbody> </table>	Value	Name	3	64x64	2	32x32	1	16x16	0	8x8										
Value	Name																				
3	64x64																				
2	32x32																				
1	16x16																				
0	8x8																				
3	<table border="1"> <tr> <td>31:3</td> <td>Reserved</td> </tr> <tr> <td></td> <td>Format: MBZ</td> </tr> <tr> <td>2</td> <td>InsertTestFlag</td> </tr> <tr> <td></td> <td>Format: U1</td> </tr> <tr> <td></td> <td> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>[Default]</td> </tr> <tr> <td>1h</td> <td></td> </tr> </tbody> </table> </td> </tr> <tr> <td></td> <td> <table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> <tr> <td> <p>CABAC 0 Word Insertion Test Enable (Encoder Only) This bit will modify CABAC K equation so that a positive K value can be generated easily. This is done for validation purpose only. In normal usage this bit should be set to 0.</p> <p>Regular equation for generating 'K' value when CABAC 0 Word Insertion Test Enable is set to 0.</p> $K = \{ [((96 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Modified equation when CABAC 0 Word Insertion Test Enable bit set to 1.</p> $K = \{ [((1536 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Encoder only feature.</p> <p>This bit should be set to 0 in regular PAK mode.</p> </td> </tr> </table> </td> </tr> </table>	31:3	Reserved		Format: MBZ	2	InsertTestFlag		Format: U1		<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>[Default]</td> </tr> <tr> <td>1h</td> <td></td> </tr> </tbody> </table>	Value	Name	0h	[Default]	1h			<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> <tr> <td> <p>CABAC 0 Word Insertion Test Enable (Encoder Only) This bit will modify CABAC K equation so that a positive K value can be generated easily. This is done for validation purpose only. In normal usage this bit should be set to 0.</p> <p>Regular equation for generating 'K' value when CABAC 0 Word Insertion Test Enable is set to 0.</p> $K = \{ [((96 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Modified equation when CABAC 0 Word Insertion Test Enable bit set to 1.</p> $K = \{ [((1536 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Encoder only feature.</p> <p>This bit should be set to 0 in regular PAK mode.</p> </td> </tr> </table>	Programming Notes	<p>CABAC 0 Word Insertion Test Enable (Encoder Only) This bit will modify CABAC K equation so that a positive K value can be generated easily. This is done for validation purpose only. In normal usage this bit should be set to 0.</p> <p>Regular equation for generating 'K' value when CABAC 0 Word Insertion Test Enable is set to 0.</p> $K = \{ [((96 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Modified equation when CABAC 0 Word Insertion Test Enable bit set to 1.</p> $K = \{ [((1536 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Encoder only feature.</p> <p>This bit should be set to 0 in regular PAK mode.</p>
31:3	Reserved																				
	Format: MBZ																				
2	InsertTestFlag																				
	Format: U1																				
	<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>[Default]</td> </tr> <tr> <td>1h</td> <td></td> </tr> </tbody> </table>	Value	Name	0h	[Default]	1h															
Value	Name																				
0h	[Default]																				
1h																					
	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> <tr> <td> <p>CABAC 0 Word Insertion Test Enable (Encoder Only) This bit will modify CABAC K equation so that a positive K value can be generated easily. This is done for validation purpose only. In normal usage this bit should be set to 0.</p> <p>Regular equation for generating 'K' value when CABAC 0 Word Insertion Test Enable is set to 0.</p> $K = \{ [((96 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Modified equation when CABAC 0 Word Insertion Test Enable bit set to 1.</p> $K = \{ [((1536 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Encoder only feature.</p> <p>This bit should be set to 0 in regular PAK mode.</p> </td> </tr> </table>	Programming Notes	<p>CABAC 0 Word Insertion Test Enable (Encoder Only) This bit will modify CABAC K equation so that a positive K value can be generated easily. This is done for validation purpose only. In normal usage this bit should be set to 0.</p> <p>Regular equation for generating 'K' value when CABAC 0 Word Insertion Test Enable is set to 0.</p> $K = \{ [((96 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Modified equation when CABAC 0 Word Insertion Test Enable bit set to 1.</p> $K = \{ [((1536 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Encoder only feature.</p> <p>This bit should be set to 0 in regular PAK mode.</p>																		
Programming Notes																					
<p>CABAC 0 Word Insertion Test Enable (Encoder Only) This bit will modify CABAC K equation so that a positive K value can be generated easily. This is done for validation purpose only. In normal usage this bit should be set to 0.</p> <p>Regular equation for generating 'K' value when CABAC 0 Word Insertion Test Enable is set to 0.</p> $K = \{ [((96 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Modified equation when CABAC 0 Word Insertion Test Enable bit set to 1.</p> $K = \{ [((1536 * pic_bin_count()) - (RawMinCUBits * PicSizeInMinCUs * 3) + 1023) / 1024] - bytes_in_picture \} / 3$ <p>Encoder only feature.</p> <p>This bit should be set to 0 in regular PAK mode.</p>																					



HCP_PIC_STATE												
1	<p>CurPicIsl</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Specifies that the current picture is comprised solely of I slices and that there are no P or B slices in the picture.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Current picture has at least one P or B slice</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This bit should be set to "0". Note: The value of "1" setting ("Current picture is comprised solely of I slices") is REMOVED. This bit is used for hardware optimization only. There is not enough information to set this bit to "1" correctly.</p>	Format:	U1	Value	Name	0	Current picture has at least one P or B slice					
	Format:	U1										
Value	Name											
0	Current picture has at least one P or B slice											
0	<p>ColPicIsl</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Specifies that the collocated picture is comprised solely of I slices and that there are no P or B slices in the picture.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Collocated picture has at least one P or B slice</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This bit should be set to "0". Note: The value of "1" setting ("Collocated picture is comprised solely of I slices") is REMOVED. This bit is used for hardware optimization only. There is not enough information to set this bit to "1" correctly.</p>	Format:	U1	Value	Name	0	Collocated picture has at least one P or B slice					
	Format:	U1										
Value	Name											
0	Collocated picture has at least one P or B slice											
4	<p>31:28 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ										
	<p>27 CU packet structure</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th colspan="3" style="background-color: #e1eef6;">Description</th> </tr> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 75%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>VME</td> <td>CU packet structure is in output format of HW VME which is 1/2 CL per CU</td> </tr> <tr> <td>1</td> <td>ExtEnc</td> <td>CU packet structure is in output format of non HW VME (External Enc/SW) which is 1 CL per CU</td> </tr> </tbody> </table>	Description			Value	Name	Description	0	VME	CU packet structure is in output format of HW VME which is 1/2 CL per CU	1	ExtEnc
Description												
Value	Name	Description										
0	VME	CU packet structure is in output format of HW VME which is 1/2 CL per CU										
1	ExtEnc	CU packet structure is in output format of non HW VME (External Enc/SW) which is 1 CL per CU										
<p>26 strong_intra_smoothing_enable_flag</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table>	Format:	U1										
Format:	U1											



HCP_PIC_STATE												
	25	transquant_bypass_enable_flag										
	Format: _____ Enable											
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>cu_transquant_bypass is not supported</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>cu_transquant_bypass is supported</td> </tr> </tbody> </table>			Value	Name	Description	0	Disable	cu_transquant_bypass is not supported	1	Enable	cu_transquant_bypass is supported
	Value	Name	Description									
	0	Disable	cu_transquant_bypass is not supported									
	1	Enable	cu_transquant_bypass is supported									
	24	Reserved										
	Format: _____ MBZ											
	23	amp_enabled_flag										
	Format: _____ Enable											
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>Asymmetric motion partitions cannot be used in coding tree blocks.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Support asymmetric motion partitions, i.e. PartMode equal to PART_2NxN_U, PART_2NxN_D, PART_nLx2N, or PART_nRx2N.</td> </tr> </tbody> </table>			Value	Name	Description	0	Disable	Asymmetric motion partitions cannot be used in coding tree blocks.	1	Enable	Support asymmetric motion partitions, i.e. PartMode equal to PART_2NxN _U , PART_2NxN _D , PART_nLx2N, or PART_nRx2N.
	Value	Name	Description									
	0	Disable	Asymmetric motion partitions cannot be used in coding tree blocks.									
	1	Enable	Support asymmetric motion partitions, i.e. PartMode equal to PART_2NxN _U , PART_2NxN _D , PART_nLx2N, or PART_nRx2N.									
	22	transform_skip_enabled_flag										
	Format: _____ Enable											
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>transform_skip_flag is not supported in the residual coding</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>transform_skip_flag is supported</td> </tr> </tbody> </table>			Value	Name	Description	0	Disable	transform_skip_flag is not supported in the residual coding	1	Enable	transform_skip_flag is supported
	Value	Name	Description									
	0	Disable	transform_skip_flag is not supported in the residual coding									
	1	Enable	transform_skip_flag is supported									
	21	BottomField										
	Format: _____ U1											
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Bottom Field</td> <td></td> </tr> <tr> <td>1</td> <td>Top Field</td> <td></td> </tr> </tbody> </table>			Value	Name	Description	0	Bottom Field		1	Top Field	
	Value	Name	Description									
	0	Bottom Field										
1	Top Field											
Programming Notes												
Must be zero for encoder only												
20	FieldPic											
Format: _____ U1												
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Video Frame</td> <td></td> </tr> <tr> <td>1</td> <td>Video Field</td> <td></td> </tr> </tbody> </table>			Value	Name	Description	0	Video Frame		1	Video Field		
Value	Name	Description										
0	Video Frame											
1	Video Field											
Programming Notes												



HCP_PIC_STATE											
	Must be zero for encoder only.										
19	weighted_pred_flag	Format: U1									
18	weighted_bipred_flag	Format: U1									
17	tiles_enabled_flag	Format: U1									
Programming Notes											
Tiling is not supported and this bit should be set to 0.											
16	entropy_coding_sync_enabled_flag	Format: U1									
Not used in encoder mode											
15	Reserved										
15	loop_filter_across_tiles_enabled_flag	Format: U1									
14	Reserved	Format: MBZ									
13	sign_data_hiding_flag	Format: Enable									
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>Specifies that sign bit hiding is disabled.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Specifies that sign bit hiding is enabled.</td> </tr> </tbody> </table>			Value	Name	Description	0	Disable	Specifies that sign bit hiding is disabled.	1	Enable	Specifies that sign bit hiding is enabled.
Value	Name	Description									
0	Disable	Specifies that sign bit hiding is disabled.									
1	Enable	Specifies that sign bit hiding is enabled.									
Programming Notes											
Currently not supported in encoder, so must be set to 0 for encoding session.											
12:10	log2_parallel_merge_level_minus2	Format: U3									
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>[0,4]</td> <td>Valid Range</td> <td>The value of log2_parallel_merge_level_minus2 shall be in the range of 0 to Log2CtbSizeYCbLog2SizeY - 2, inclusive.</td> </tr> </tbody> </table>			Value	Name	Programming Notes	[0,4]	Valid Range	The value of log2_parallel_merge_level_minus2 shall be in the range of 0 to Log2CtbSizeYCbLog2SizeY - 2, inclusive.			
Value	Name	Programming Notes									
[0,4]	Valid Range	The value of log2_parallel_merge_level_minus2 shall be in the range of 0 to Log2CtbSizeYCbLog2SizeY - 2, inclusive.									
Programming Notes											
For encoder, always set to 0 (Intel restriction).											



HCP_PIC_STATE											
9	constrained_intra_pred_flag	Format: U1									
	pcm_loop_filter_disable_flag	Format: U1									
	7:6	diff_cu_qp_delta_depth (or named as max_dqp_depth)									
		Format: U2									
	Programming Notes										
	cu_qp_delta_enabled_flag = 1 and Max_DQP_Level = 0 or 3 is supported in PAK standalone mode.										
	cu_qp_delta_enabled_flag/max_dqp_depth: 1/0: has cu qp delta. (cu depth <= max_dqp_depth) will have cu qp delta coded. Only allow QP change across Slices, no change across LCU or CU.										
	5	cu_qp_delta_enabled_flag	Format: U1								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">Disable</td> <td>Does not allow QP change at CU or LCU level, the same QP is used for the entire slice. Max_DQP_Level = 0 (i.e. diff_cu_qp_delta_depath = 0).</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">Enable</td> <td>Allow QP change at CU level. MAX_DQP_level can be >0.</td> </tr> </tbody> </table>		Value	Name	Description	0	Disable	Does not allow QP change at CU or LCU level, the same QP is used for the entire slice. Max_DQP_Level = 0 (i.e. diff_cu_qp_delta_depath = 0).	1	Enable	Allow QP change at CU level. MAX_DQP_level can be >0.
Value	Name	Description									
0	Disable	Does not allow QP change at CU or LCU level, the same QP is used for the entire slice. Max_DQP_Level = 0 (i.e. diff_cu_qp_delta_depath = 0).									
1	Enable	Allow QP change at CU level. MAX_DQP_level can be >0.									
Programming Notes											
cu_qp_delta_enabled_flag = 1 and Max_DQP_Level = 0 or 3 is supported for PAK standalone mode.											
4	pcm_enabled_flag	Format: U1									
3	sample_adaptive_offset_enabled_flag	Format: U1									
Programming Notes											
HW does not support SAO filtering in encoder mode											
2:0	Reserved	Format: MBZ									
5	31:30	Reserved Format: MBZ									



HCP_PIC_STATE																						
	29:27	<p>bit_depth_luma_minus8</p> <table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>This specifies the number of bit allow for Luma pixels. In 8 bit mode, this must be set to 0. Encoder: Supports bit depths 8, 10 and 12 only. Encoder: Does not support 10 or 12 bit Source Pixels and 8bit PAK i.e. the source pixel depth should be less than or equal to PAK bit depth.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>luma_8bit</td> <td></td> </tr> <tr> <td>1</td> <td>luma_9bit</td> <td>Only HEVC decoder supports 9 bits luma. HEVC encoder does not supports 9 bits luma.</td> </tr> <tr> <td>2</td> <td>luma_10bit</td> <td></td> </tr> <tr> <td>3</td> <td>luma_11bit</td> <td></td> </tr> <tr> <td>4</td> <td>luma_12bit</td> <td></td> </tr> </tbody> </table>	Format:	U3	Value	Name	Description	0	luma_8bit		1	luma_9bit	Only HEVC decoder supports 9 bits luma. HEVC encoder does not supports 9 bits luma.	2	luma_10bit		3	luma_11bit		4	luma_12bit	
	Format:	U3																				
	Value	Name	Description																			
	0	luma_8bit																				
	1	luma_9bit	Only HEVC decoder supports 9 bits luma. HEVC encoder does not supports 9 bits luma.																			
	2	luma_10bit																				
	3	luma_11bit																				
	4	luma_12bit																				
	26:24	<p>bit_depth_chroma_minus8</p> <table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>This specifies the number of bit allow for Chroma pixels. In 8 bit mode, this must be set to 0. Encoder: Supports bit depths 8, 10 and 12 only. And also it must be same as Luma. Encoder: Does not support 10 or 12 bit Source Pixels and 8bit PAK. i.e. The source pixel depth should be less than or equal to the PAK bit depth.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>chroma_8bit</td> <td></td> </tr> <tr> <td>1</td> <td>chroma_9bit</td> <td>Only HEVC decoder supports 9 bits chroma. HEVC encoder does not supports 9 bits chroma.</td> </tr> <tr> <td>2</td> <td>chroma_10bit</td> <td></td> </tr> <tr> <td>3</td> <td>chroma_11bit</td> <td></td> </tr> <tr> <td>4</td> <td>chroma_12bit</td> <td></td> </tr> </tbody> </table>	Format:	U3	Value	Name	Description	0	chroma_8bit		1	chroma_9bit	Only HEVC decoder supports 9 bits chroma. HEVC encoder does not supports 9 bits chroma.	2	chroma_10bit		3	chroma_11bit		4	chroma_12bit	
	Format:	U3																				
	Value	Name	Description																			
	0	chroma_8bit																				
1	chroma_9bit	Only HEVC decoder supports 9 bits chroma. HEVC encoder does not supports 9 bits chroma.																				
2	chroma_10bit																					
3	chroma_11bit																					
4	chroma_12bit																					
23:20	<p>pcm_sample_bit_depth_luma_minus1</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <table border="1"> <thead> <tr> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Encoder support bitdepth 8 only. Must be same for luma and chroma Cb and Cr.</td> </tr> </tbody> </table>	Format:	U4	Description	Encoder support bitdepth 8 only. Must be same for luma and chroma Cb and Cr.																	
Format:	U4																					
Description																						
Encoder support bitdepth 8 only. Must be same for luma and chroma Cb and Cr.																						
19:16	<p>pcm_sample_bit_depth_chroma_minus1</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <table border="1"> <thead> <tr> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Encoder support bitdepth 8 only. Must be same for luma and chroma Cb and Cr.</td> </tr> </tbody> </table>	Format:	U4	Description	Encoder support bitdepth 8 only. Must be same for luma and chroma Cb and Cr.																	
Format:	U4																					
Description																						
Encoder support bitdepth 8 only. Must be same for luma and chroma Cb and Cr.																						



HCP_PIC_STATE															
	<p>15:13 max_transform_hierarchy_depth_inter(or named as tu_max_depth_inter)</p> <table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Maximum TU split depths for inter blocks</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">For encoder, always set to 2 to allow max 2 levels of split. For more splitting rely on CU split to match the content (Intel restriction)</td> </tr> </table>	Format:	U3	Programming Notes		For encoder, always set to 2 to allow max 2 levels of split. For more splitting rely on CU split to match the content (Intel restriction)									
	Format:	U3													
	Programming Notes														
	For encoder, always set to 2 to allow max 2 levels of split. For more splitting rely on CU split to match the content (Intel restriction)														
<p>12:10 max_transform_hierarchy_depth_intra (or named as tu_max_depth_intra)</p> <table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Maximum TU split depth for intra blocks.</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">For encoder, always set to 2 to allow max 2 levels of split. For more splitting, rely on CU split to match the content (Intel restriction).</td> </tr> </table>	Format:	U3	Programming Notes		For encoder, always set to 2 to allow max 2 levels of split. For more splitting, rely on CU split to match the content (Intel restriction).										
Format:	U3														
Programming Notes															
For encoder, always set to 2 to allow max 2 levels of split. For more splitting, rely on CU split to match the content (Intel restriction).															
<p>9:5 pic_cr_qp_offset</p> <table border="1"> <tr> <td>Format:</td> <td>S4</td> </tr> </table> <p>Valid range -12 to +12</p>	Format:	S4													
Format:	S4														
<p>4:0 pic_cb_qp_offset</p> <table border="1"> <tr> <td>Format:</td> <td>S4</td> </tr> </table> <p>Valid range -12 to +12</p>	Format:	S4													
Format:	S4														
6	<p>31:30 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ													
<p>29 Load Slice Pointer Flag</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>LoadBitStreamPointerPerSlice (Encoder-only)</p> <p>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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>Load BitStream Pointer only once for the first slice of a frame.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Load/reload BitStream Pointer only once for the each slice, reload the start location of the bitstream buffer from the Indirect PAK-BSE Object Data Start Address field.</td> </tr> </tbody> </table> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">Must be zero for encoder</td> </tr> </table>	Format:	Enable	Value	Name	Description	0	Disable	Load BitStream Pointer only once for the first slice of a frame.	1	Enable	Load/reload BitStream Pointer only once for the each slice, reload the start location of the bitstream buffer from the Indirect PAK-BSE Object Data Start Address field.	Programming Notes		Must be zero for encoder	
Format:	Enable														
Value	Name	Description													
0	Disable	Load BitStream Pointer only once for the first slice of a frame.													
1	Enable	Load/reload BitStream Pointer only once for the each slice, reload the start location of the bitstream buffer from the Indirect PAK-BSE Object Data Start Address field.													
Programming Notes															
Must be zero for encoder															



HCP_PIC_STATE		
28:27	Reserved	
	Format:	MBZ
26	FrameSzUnderStatusEn - FrameBitRateMinReportMask	
	Format:	Enable
	This is a mask bit controlling if the condition of frame level bit count is less than FrameBitRateMin.	
	Value	Name
0	Disable	Do not update bit 2 (Frame Bit Count Violate -- under run) of HCP_IMAGE_STATUS control register. NOTE: This bit MUST BE set to zero for the last BRC pass if SAO first pass also enabled. This will ensure that HW picks up right accumulated deltaQP for SAO second pass.
1	Enable	Set bit 2 (Frame Bit Count Violate -- under run) of HCP_IMAGE_STATUS control register if the total frame level bit counter is less than or equal to Frame Bit Rate Minimum limit. HW does not use this bit to set the bit in HCP_IMAGE_STATUS_CONTROL register. It's used pass the bit in HCP_IMAGE_STATUS_MASK register
25	FrameSzOverStatusEn - FrameBitRateMaxReportMask	
	Format:	Enable
	This is a mask bit controlling if the condition of frame level bit count exceeds FrameBitRateMax.	
	Value	Name
0	Disable	Do not update bit 1 of HCP_IMAGE_STATUS control register. NOTE: This bit MUST BE set to zero for the last BRC pass if SAO first pass also enabled. This will ensure that HW picks up right accumulated deltaQP for SAO second pass.
1	Enable	Set bit 1 of HCP_IMAGE_STATUS control register if the total frame level bit counter is greater than or equal to Frame Bit Rate Maximum limit. HW does not use this bit to set the bit in HCP_IMAGE_STATUS_CONTROL register. It's used pass the bit in HCP_IMAGE_STATUS_MASK register
24	LCUMaxBitStatusEn - LCUMaxSizeReportMask	
	Format:	Enable
	This is a mask bit controlling if the condition of any LCU in the frame exceeds LCUMaxSize.	
	Value	Name
0	Disable	Do not update bit 0 of HCP_IMAGE_STATUS control register. NOTE: This bit MUST BE set to zero for the last BRC pass if SAO first pass also enabled. This will ensure that HW picks up right accumulated



HCP_PIC_STATE													
		<p>deltaQP for SAO sencond pass.</p> <table border="1"> <tr> <td style="width: 50px;">1</td> <td style="width: 100px;">Enable</td> <td>Set bit 0 of HCP_IMAGE_STATUS control register if the total bit counter for the current LCU is greater than the LCU Conformance Max size limit. HW does not use this bit to set the bit in HCP_IMAGE_STATUS_CONTROL register.</td> </tr> </table>	1	Enable	Set bit 0 of HCP_IMAGE_STATUS control register if the total bit counter for the current LCU is greater than the LCU Conformance Max size limit. HW does not use this bit to set the bit in HCP_IMAGE_STATUS_CONTROL register.								
1	Enable	Set bit 0 of HCP_IMAGE_STATUS control register if the total bit counter for the current LCU is greater than the LCU Conformance Max size limit. HW does not use this bit to set the bit in HCP_IMAGE_STATUS_CONTROL register.											
Programming Notes													
Encoder Only													
23:17	Reserved	<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												
16	NonFirstPassFlag	<table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This signals the current pass is not the first pass. It will imply designate HW behavior.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>If it is initial-Pass, this bit is set to 0.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>For subsequent passes, this bit is set to 1.</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0	Disable	If it is initial-Pass, this bit is set to 0.	1	Enable	For subsequent passes, this bit is set to 1.
Format:	Enable												
Value	Name	Description											
0	Disable	If it is initial-Pass, this bit is set to 0.											
1	Enable	For subsequent passes, this bit is set to 1.											
Programming Notes													
Encoder Only													
15:0	LCU Max BitSize Allowed	<table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>This field specifies the Max LCU Bit Size allowed in according to the spec conformance. However, driver can program a different value from the spec for other encoding purpose.</p> <p>If LCU Limit exceeds, LCUMaxBitStatus in MMIO would be set to 1. If LCU Limit exceeds, LCUMaxBitStatus in MMIO would be set to 1 and also in CU. Statistics Streamout, LCU Limit exceed field would be set to 1 at the last CU of a LCU.</p>	Format:	U16									
Format:	U16												
Programming Notes													
Encoder Only													
7 Programming Notes: Encoder Only	31	FrameBitrateMaxUnit											
		<table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>This field is the Frame Bitrate Maximum Limit Units.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Byte</td> <td>32byte unit</td> </tr> <tr> <td>1</td> <td>Kilo Byte</td> <td>4kbyte unit</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	0	Byte	32byte unit	1	Kilo Byte	4kbyte unit
	Format:	U1											
	Value	Name	Description										
0	Byte	32byte unit											
1	Kilo Byte	4kbyte unit											



HCP_PIC_STATE											
		Programming Notes									
		Encoder Only									
	30:14	<p>Reserved</p> <p>Format: MBZ</p> <p>Reserved for future expansion of the Max Rate.</p>									
	13:0	<p>FrameBitRateMax</p> <p>Format: U14</p> <p>This field is the Frame Bitrate Maximum Limit. This field along with FrameBitRateMaxUnit determines maximum allowed bits in a frame before multi-pass gets triggered (when enabled). In other words, multi-pass is triggered when the actual frame byte count exceeds this value.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-512KB</td> <td></td> <td>The programmable range is 0-512KB when FrameBitRateMaxUnit is 0.</td> </tr> <tr> <td>0-64MB</td> <td></td> <td>The programmable range is 0-64Mbyte when FrameBitRateMaxUnit is 1.</td> </tr> </tbody> </table>	Value	Name	Description	0-512KB		The programmable range is 0-512KB when FrameBitRateMaxUnit is 0.	0-64MB		The programmable range is 0-64Mbyte when FrameBitRateMaxUnit is 1.
Value	Name	Description									
0-512KB		The programmable range is 0-512KB when FrameBitRateMaxUnit is 0.									
0-64MB		The programmable range is 0-64Mbyte when FrameBitRateMaxUnit is 1.									
8	31	<p>FrameBitRateMinUnit</p> <p>Format: U1</p> <p>This field is the Frame Bitrate Minimum Limit Units.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Byte</td> <td>32byte unit</td> </tr> <tr> <td>1</td> <td>Kilo Byte</td> <td>4kbyte unit</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Encoder Only</p>	Value	Name	Description	0	Byte	32byte unit	1	Kilo Byte	4kbyte unit
Value	Name	Description									
0	Byte	32byte unit									
1	Kilo Byte	4kbyte unit									
	30:14	<p>Reserved</p> <p>Format: MBZ</p> <p>Reserved for future expansion of the Min Rate.</p>									
	13:0	<p>FrameBitRateMin</p> <p>Format: U14</p> <p>This field is the Frame Bitrate Minimum Limit. This field along with FrameBitRateMinUnit determines minimum allowed bits in a frame before multi-pass gets triggered (when enabled). In other words, multi-pass is triggered when the actual frame byte count happen to be below this value.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-512KB</td> <td></td> <td>The programmable range is 0-512KB when FrameBitRateMinUnit is 0.</td> </tr> </tbody> </table>	Value	Name	Description	0-512KB		The programmable range is 0-512KB when FrameBitRateMinUnit is 0.			
Value	Name	Description									
0-512KB		The programmable range is 0-512KB when FrameBitRateMinUnit is 0.									



HCP_PIC_STATE																		
		<table border="1"> <tr> <td>0-64MB</td> <td></td> <td>The programmable range is 0-64Mbyte when FrameBitrateMinUnit is 1.</td> </tr> <tr> <td colspan="3" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="3">Encoder Only</td> </tr> </table>	0-64MB		The programmable range is 0-64Mbyte when FrameBitrateMinUnit is 1.	Programming Notes			Encoder Only									
0-64MB		The programmable range is 0-64Mbyte when FrameBitrateMinUnit is 1.																
Programming Notes																		
Encoder Only																		
9	31	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
	Format:	MBZ																
	30:16	<p>FrameBitRateMaxDelta</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Always</td> </tr> <tr> <td>Format:</td> <td>U15</td> </tr> </table> <p>This field is used to select the slice delta QP when FrameBitRateMax is exceeded. It shares the same FrameBitrateMaxUnit.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>[Default]</td> <td></td> </tr> <tr> <td>0-1024KB</td> <td></td> <td>The Programmable range is 0-1024KB when FrameBitRateMaxUnit is 0.</td> </tr> <tr> <td>0-128MB</td> <td></td> <td>The Programmable range is 0-128MB when FrameBitRateMaxUnit is 1.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Encoder Only</p>	Exists If:	//Always	Format:	U15	Value	Name	Description	0	[Default]		0-1024KB		The Programmable range is 0-1024KB when FrameBitRateMaxUnit is 0.	0-128MB		The Programmable range is 0-128MB when FrameBitRateMaxUnit is 1.
	Exists If:	//Always																
	Format:	U15																
Value	Name	Description																
0	[Default]																	
0-1024KB		The Programmable range is 0-1024KB when FrameBitRateMaxUnit is 0.																
0-128MB		The Programmable range is 0-128MB when FrameBitRateMaxUnit is 1.																
15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																	
14:0	<p>FrameBitRateMinDelta</p> <table border="1"> <tr> <td>Format:</td> <td>U15</td> </tr> </table> <p>This field is used to select the slice delta QP when FrameBitRateMin is exceeded. It shares the same FrameBitrateMinUnit.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>[Default]</td> <td></td> </tr> <tr> <td>0-1024KB</td> <td></td> <td>The Programmable range is 0-1024KB when FrameBitRateMinUnit is 0.</td> </tr> <tr> <td>0-128MB</td> <td></td> <td>The Programmable range is 0-128MB when FrameBitRateMinUnit is 1.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>HW requires the following condition: FrameBitRateMinDelta <= 2*FrameBitRateMinMust be true, otherwise it may cause unpredicted behavior.</p> <p>Encoder Only</p>	Format:	U15	Value	Name	Description	0	[Default]		0-1024KB		The Programmable range is 0-1024KB when FrameBitRateMinUnit is 0.	0-128MB		The Programmable range is 0-128MB when FrameBitRateMinUnit is 1.			
Format:	U15																	
Value	Name	Description																
0	[Default]																	
0-1024KB		The Programmable range is 0-1024KB when FrameBitRateMinUnit is 0.																
0-128MB		The Programmable range is 0-128MB when FrameBitRateMinUnit is 1.																



HCP_PIC_STATE				
10..11	63:0	<p>FrameDeltaQpMax</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">FrameDeltaQp</td> </tr> </table> <p>Range: [0:MAX_QP_DELTA]</p> <p>Frame level delta QP which should be used in case FrameSize - FrameBitRateMax in the range of $((\Delta QpMaxRange[n] * FrameBitRateMaxDelta \gg 5))$, $\Delta QpMaxRange[n+1] * FrameBitRateMaxDelta \gg 5$.</p> <p style="text-align: center;">Programming Notes</p> <p>If $n == 7$, DeltaQpMaxRange is infinity. Format: 8 bit sign-magnitude, Range: [0:63]</p> <p>Encoder Only</p>	Format:	FrameDeltaQp
Format:	FrameDeltaQp			
12..13	63:0	<p>FrameDeltaQpMin</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">FrameDeltaQp</td> </tr> </table> <p>Range: [0:MIN_QP_DELTA]</p> <p>Frame level delta QP which should be used in case FrameSize - FrameBitRateMin in the range of $((\Delta QpMinRange[n] * FrameBitRateMinDelta \gg 5))$, $\Delta QpMinRange[n+1] * FrameBitRateMinDelta \gg 5$.</p> <p style="text-align: center;">Programming Notes</p> <p>If $n == 7$, DeltaQpMinRange is infinity. Format: 8 bit sign-magnitude Range: [-63:0]</p> <p>Encoder Only</p>	Format:	FrameDeltaQp
Format:	FrameDeltaQp			
14..15	63:0	<p>FrameDeltaQpMaxRange</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">FrameDeltaQpRange</td> </tr> </table> <p>Range: [0:U8_MAX]</p> <p>Condition: $FrameDeltaQpMaxRange[n] \geq FrameDeltaQpMaxRange[n-1]$</p> <p>This field is to calculate ranges for Frame level delta QP, specifically Frame level delta QP[n] and Frame level delta QP[n+1].</p> <p style="text-align: center;">Programming Notes</p> <p>If $n == 0$, FrameDeltaQpMaxRange is zero.</p> <p>Encoder Only</p>	Format:	FrameDeltaQpRange
Format:	FrameDeltaQpRange			
16..17	63:0	<p>FrameDeltaQpMinRange</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">FrameDeltaQpRange</td> </tr> </table> <p>Range: [0:U8_MAX]</p>	Format:	FrameDeltaQpRange
Format:	FrameDeltaQpRange			



HCP_PIC_STATE																	
		<p>Condition: $\text{FrameDeltaQpMinRange}[n] \geq \text{FrameDeltaQpMinRange}[n-1]$</p> <p>This field is to calculate ranges for Frame level delta QP, specifically Frame level delta QP[n] and Frame level delta QP[n+1].</p> <p style="text-align: center;">Programming Notes</p> <p>If $n == 0$, FrameDeltaQpMinRange is zero.</p> <p>Encoder Only</p>															
18	31:30	<p>MinFrameSizeUnits</p> <p>Format: U2</p> <p>This field is the Minimum Frame Size Units</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>4Kb</td> <td>Minimum Frame Size is in 4Kbytes.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>16Kb</td> <td>Minimum Frame Size is in 16Kbytes.</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Reserved</td> <td></td> </tr> <tr> <td style="text-align: center;">3</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Encoder Only</p>	Value	Name	Description	0	4Kb	Minimum Frame Size is in 4Kbytes.	1	16Kb	Minimum Frame Size is in 16Kbytes.	2	Reserved		3	Reserved	
Value	Name	Description															
0	4Kb	Minimum Frame Size is in 4Kbytes.															
1	16Kb	Minimum Frame Size is in 16Kbytes.															
2	Reserved																
3	Reserved																
	29:16	<p>Reserved</p> <p>Format: MBZ</p>															
	15:0	<p>MinFrameSize</p> <p>Default Value: 0</p> <p>Format: U16</p> <p>Minimum Frame Size [15:0] (in Word, 16-bit)(Encoder Only)</p> <p>Minimum Frame Size is specified to compensate for intel Rate Control Currently zero fill (no need to perform emulation byte insertion) is done only to the end of the CABAC_ZERO_WORD insertion (if any) at the last slice of a picture. It is needed for CBR. Intel encoder parameter, not part of DXVA. The caller should always make sure that the value, represented by Minimum Frame Size, is always less than maximum frame size FrameBitRateMax. This field is reserved in Decode mode.</p> <p style="text-align: center;">Programming Notes</p> <p>Programmable range is $0..(2^{16}-1) * 2^{12}$ when MinFrameSizeUnits is 0. (4KB unit)</p> <p>Programmable range is $0..(2^{16}-1) * 2^{14}$ when MinFrameSizeUnits is 1. (16KB unit)</p> <p>Encoder Only</p>															
19..31	31:0	<p>Reserved</p> <p>Format: MBZ</p>															



HCP_PIPE_BUF_ADDR_STATE

HCP_PIPE_BUF_ADDR_STATE				
Source:	VideoCS			
Length Bias:	2			
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>This state command provides the memory base addresses for the row store buffer and reconstructed picture output buffers required by the HCP.</p> <p>This is a picture level state command and is shared by both encoding and decoding processes.</p>				
Programming Notes				
All pixel surface addresses must be 4K byte aligned. There is a max of 8 Reference Picture Buffer Addresses, and all share the same third address DW in specifying 48-bit address.				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	3h PARALLEL_VIDEO_PIPE	
		Format:	OpCode	
	28:27	28:27	Pipeline Type	
			Default Value:	2h
			Format:	OpCode
	26:23	26:23	Media Instruction Opcode	
			Default Value:	7h Codec/Engine Name
			Format:	OpCode
			Codec/Engine Name = HCP = 7h	
	22:16	22:16	Media Instruction Command	
			Default Value:	2h HCP_PIPE_BUF_ADDR_STATE
			Format:	OpCode
	15:12	15:12	Reserved	
Format:			MBZ	
11:0	11:0	Dword Length		
		Format:	=n	
		(Excludes Dwords 0, 1).		
		Value	Name	
		51h		
66h				



HCP_PIPE_BUF_ADDR_STATE				
1..2	63:0	<p>Decoded Picture</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>SplitBaseAddress4KByteAligned</td> </tr> </table> <p>Frame buffer address for the final decoded picture YUV output.</p>	Format:	SplitBaseAddress4KByteAligned
Format:	SplitBaseAddress4KByteAligned			
3	31:0	<p>Decoded Picture</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MemoryAddressAttributes</td> </tr> </table>	Format:	MemoryAddressAttributes
Format:	MemoryAddressAttributes			
4..5	63:0	<p>Deblocking Filter Line Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>SplitBaseAddress64ByteAligned</td> </tr> </table> <p>Base address of the filter line buffer (read/write) used by the Deblocking Filter.</p>	Format:	SplitBaseAddress64ByteAligned
Format:	SplitBaseAddress64ByteAligned			
6	31:0	<p>Deblocking Filter Line Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MemoryAddressAttributes</td> </tr> </table>	Format:	MemoryAddressAttributes
Format:	MemoryAddressAttributes			
7..8	63:0	<p>Deblocking Filter Tile Line Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>SplitBaseAddress64ByteAligned</td> </tr> </table> <p>Base address of the tile line buffer (read/write) used by the Deblocking Filter.</p>	Format:	SplitBaseAddress64ByteAligned
Format:	SplitBaseAddress64ByteAligned			
9	31:0	<p>Deblocking Filter Tile Line Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MemoryAddressAttributes</td> </tr> </table>	Format:	MemoryAddressAttributes
Format:	MemoryAddressAttributes			
10..11	63:0	<p>Deblocking Filter Tile Column Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>SplitBaseAddress64ByteAligned</td> </tr> </table> <p>Base address of the tile column buffer (read/write) used by the Deblocking Filter.</p>	Format:	SplitBaseAddress64ByteAligned
Format:	SplitBaseAddress64ByteAligned			
12	31:0	<p>Deblocking Filter Tile Column Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MemoryAddressAttributes</td> </tr> </table>	Format:	MemoryAddressAttributes
Format:	MemoryAddressAttributes			
13..14	63:0	<p>Metadata Line Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>SplitBaseAddress64ByteAligned</td> </tr> </table> <p>Base address for the Metadata Line buffer.</p>	Format:	SplitBaseAddress64ByteAligned
Format:	SplitBaseAddress64ByteAligned			
15	31:0	<p>Metadata Line Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MemoryAddressAttributes</td> </tr> </table>	Format:	MemoryAddressAttributes
Format:	MemoryAddressAttributes			
16..17	63:0	<p>Metadata Tile Line Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>SplitBaseAddress64ByteAligned</td> </tr> </table> <p>Base address for the Metadata Tile Line buffer.</p>	Format:	SplitBaseAddress64ByteAligned
Format:	SplitBaseAddress64ByteAligned			
18	31:0	<p>Metadata Tile Line Buffer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MemoryAddressAttributes</td> </tr> </table>	Format:	MemoryAddressAttributes
Format:	MemoryAddressAttributes			



HCP_PIPE_BUF_ADDR_STATE		
19..20	63:0	Metadata Tile Column Buffer Format: SplitBaseAddress64ByteAligned Base address for the Metadata Tile Column buffer.
21	31:0	Metadata Tile Column Buffer Format: MemoryAddressAttributes
22..23	63:0	SAO Line Buffer Format: SplitBaseAddress64ByteAligned Base address for the SAO Line buffer.
24	31:0	SAO Line Buffer Format: MemoryAddressAttributes
25..26	63:0	SAO Tile Line Buffer Format: SplitBaseAddress64ByteAligned Base address for the SAO Tile Line buffer.
27	31:0	SAO Tile Line Buffer Format: MemoryAddressAttributes
28..29	63:0	SAO Tile Column Buffer Format: SplitBaseAddress64ByteAligned Base address for the SAO Tile Column buffer.
30	31:0	SAO Tile Column Buffer Format: MemoryAddressAttributes
31..32	63:0	Current Motion Vector Temporal Buffer Format: SplitBaseAddress64ByteAligned Base address for the Current Motion Vector Temporal buffer.
33	31:0	Current Motion Vector Temporal Buffer Format: MemoryAddressAttributes
34..35	63:0	Reserved Format: MBZ
36	31:0	Reserved Format: MBZ
37..52	63:0	Reference Picture Base Address (RefAddr[0-7]) Format: SplitBaseAddress64ByteAligned[8] Base address of the reference picture buffer.



HCP_PIPE_BUF_ADDR_STATE		
53	31:0	Reference Picture Base Address
		Format: MemoryAddressAttributes
54..55	63:0	Original Uncompressed Picture Source
		Format: SplitBaseAddress64ByteAligned
		Buffer address for fetching YUV pixel data from the original uncompressed input picture for encoding.
		This value is only valid in encoding mode.
56	31:0	Original Uncompressed Picture Source
		Format: MemoryAddressAttributes
57..58	63:0	Streamout Data Destination
		Exists If: //Decoder Only
		Format: SplitBaseAddress64ByteAligned
		Buffer address for outputting the per-block indirect data to memory when StreamOutEnable is set in the HCP_PIPE_MODE_SELECT command.
		For Decoder: this field is used for transcoding purpose.
		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.
		For Encoder: This surface is used to streamout CU records 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-block data.
59	31:0	Streamout Data Destination
		Exists If: //Decoder Only
	Format: MemoryAddressAttributes	
	31:0	Reserved
		Exists If: //Encoder Only
Format: MBZ		
60..61	63:0	Decoded Picture Status/Error Buffer Base Address or Encoded slice size streamout Base Address
		Format: SplitBaseAddress64ByteAligned
<p>Decoder Mode : Specifies the 64 byte aligned buffer address for writing a single status/error cache-line sized record into memory when the Pic Status/Error Report Enable is set in the HCP_PIPE_MODE_SELECT command. The pic status/error record is written by hardware after the picture is decoded.</p> <p>Encoder Mode: This specifies 64 byte aligned buffer address for writing Slice size, when slice size conformance is enabled.</p>		



HCP_PIPE_BUF_ADDR_STATE		
62	31:0	Decoded Picture Status/Error Buffer Base Address Format: MemoryAddressAttributes
63..64	63:0	LCU ILDB Streamout Buffer Format: SplitBaseAddress64ByteAligned Buffer address for writing ILDB parameter per LCU to memory when Deblocker Streamout Enable is set in the HCP_PIPE_MODE_SELECT Command. The ILDB MB control parameters are written by HW at the end of each reconstructed LCU. Only edge information is being streamed out.
65	31:0	LCU ILDB Streamout Buffer Format: MemoryAddressAttributes
66..81	63:0	Collocated Motion Vector Temporal Buffer[0-7] Format: SplitBaseAddress64ByteAligned[8] Base address for the Collocated Motion Vector Temporal buffer.
82	31:0	Collocated Motion Vector Temporal Buffer[0-7] Memory Address Attributes Format: MemoryAddressAttributes
83..84	63:0	VP9 Probability Buffer Read/Write Format: SplitBaseAddress64ByteAligned Specifies the 64 byte aligned buffer address for VP9 Probability Buffer. Hardware reads in the probability for decode and write out the modified probability for future frames. Driver needs to program the Initial VP9 Probability for decoding the current frame. For Key Frame, it should contain the default Key Frame Probability. For non-Key Frame, it could be a default (non-Key) or one of the 8 Reference Buffers Probability. Driver must provide a valid Initial VP9 Probability buffer.
85	31:0	VP9 Probability Buffer Read/Write Memory Address Attributes Format: MemoryAddressAttributes
86..87	63:0	VP9 Segment ID Buffer Read/Write Specifies the 64 byte aligned buffer address for VP9 SegmentID buffer. This should contain the writeout SegmentID from previous frame and will be used to predict SegmentID for the current frame. Hardware will write out SegmentID of the current frame in the same address for the next frame.
88	31:0	VP9 Segment ID buffer Read/Write Memory Address Attributes Format: MemoryAddressAttributes
89..90	63:0	VP9 HVD Line Rowstore Buffer Read/Write Format: SplitBaseAddress64ByteAligned Specifies the 64 byte aligned buffer address for HVD Tile Rowstore Buffer (bitstream decoder).



HCP_PIPE_BUF_ADDR_STATE		
91	31:0	VP9 HVD Line Rowstore buffer Read/Write Memory Address Attributes Format: MemoryAddressAttributes
92..93	63:0	VP9 HVD Tile Rowstore Buffer Read/Write Format: SplitBaseAddress64ByteAligned
94	31:0	VP9 HVD Tile Rowstore buffer Read/Write Memory Address Attributes Format: MemoryAddressAttributes
95..103	31:0	Reserved Format: MBZ



HCP_PIPE_MODE_SELECT

HCP_PIPE_MODE_SELECT		
Source:	VideoCS	
Length Bias:	2	
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The workload for the HCP is based upon a single frame decode. There are no states saved between frame decodes in the HCP. Once the bit stream DMA is configured with the HCP_BSD_OBJECT command, and the bit stream is presented to the HCP, the frame decode will begin.</p> <p>The HCP_PIPE_MODE_SELECT command is responsible for general pipeline level configuration that would normally be set once for a single stream encode or decode and would not be modified on a frame workload basis.</p> <p>This is a picture level state command and is shared by both encoding and decoding processes.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
	Format: OpCode	
	28:27	Pipeline Type
		Default Value: 2h
	Format: OpCode	
	26:23	Media Instruction Opcode
		Default Value: 7h Codec/Engine Name
		Format: OpCode
	Codec/Engine Name = HCP = 7h	
22:16	Media Instruction Command	
	Default Value: 0h HCP_PIPE_MODE_SELECT	
Format: OpCode		
15:12	Reserved	
	Format: MBZ	
11:0	DWord Length	
	Format: =n	
	(Excludes Dwords 0, 1).	
	Value	Name
	2h	Value_2
4h	Value_4	
1	31:24	Reserved



HCP_PIPE_MODE_SELECT											
23	Reserved	Format: MBZ									
22:20	Reserved	Format: MBZ									
19:18	Reserved	Format: MBZ									
17	RESERVED	Format: MBZ									
16:13	Reserved	Format: MBZ									
12	PAK Frame Level StreamOut enable	Format: U1 This bit is valid if global bitPAK Pipeline Streamout Enable is set to 1. This bit is defined to use legacy tests on HW and it's valid for both hevc/vp9. By default it should be '0'									
11	Reserved	Format: MBZ									
10	Reserved										
9	Reserved										
7:5	Codec Standard Select	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>HEVC</td> </tr> <tr> <td>1</td> <td>VP9</td> </tr> </tbody> </table>	Value	Name	0	HEVC	1	VP9			
Value	Name										
0	HEVC										
1	VP9										
4	Reserved										
3	Pic Status/Error Report Enable	Format: Enable <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>Disable status/error reporting</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Status/Error reporting is written out once per picture. The Pic Status/Error Report ID in DWord3 along with the status/error status bits are packed into one cache line and written to the Status/Error Buffer address in the HCP_PIPE_BUF_ADDR_STATE command. Must be zero for encoder mode.</td> </tr> </tbody> </table>	Value	Name	Description	0	Disable	Disable status/error reporting	1	Enable	Status/Error reporting is written out once per picture. The Pic Status/Error Report ID in DWord3 along with the status/error status bits are packed into one cache line and written to the Status/Error Buffer address in the HCP_PIPE_BUF_ADDR_STATE command. Must be zero for encoder mode.
Value	Name	Description									
0	Disable	Disable status/error reporting									
1	Enable	Status/Error reporting is written out once per picture. The Pic Status/Error Report ID in DWord3 along with the status/error status bits are packed into one cache line and written to the Status/Error Buffer address in the HCP_PIPE_BUF_ADDR_STATE command. Must be zero for encoder mode.									



HCP_PIPE_MODE_SELECT			
2	PAK Pipeline Streamout Enable		
	Format: Enable		
	Pipeline Streamout Enable is only defined for encode. It is ignored for decode.		
	Value	Name	
	0	Disable pipeline states and parameters streamout	
	1	Enable pipeline states and parameters streamout	
	Programming Notes		
	Shouldn't be turned off during SAO first PASS		
	1	Deblocker Streamout Enable	
		Format: Enable	
Deblocker Streamout Enable not currently supported for Encode or Decode			
Value		Name Description	
0		Disable Disable deblocker-only parameter streamout	
1		Enable Enable deblocker-only parameter streamout	
0	Codec Select		
	Format: U1		
	Value	Name	
	0	Decode	
	1	Encode	
2	31:0	Media Soft-Reset Counter (per 1000 clocks)	
		Format: U32	
		In decoder modes, this counter value specifies the number of clocks (per 1000) of GAC inactivity before a media soft-reset is applied to the HCP. If counter value is set to 0, the media soft-reset feature is disabled and no reset will occur. In encoder modes, this counter must be set to 0 to disable media soft reset. This feature is not supported for the encoder.	
		Value Name	
	0	Disable	
3	31:0	Pic Status/Error Report ID	
		Format: U32	
		The Pic Status/Error Report ID is a unique 32-bit unsigned integer assigned to each picture status/error output. Must be zero for encoder mode.	
		Value Name Description	
		0	32-bit unsigned
	1	Reserved	



HCP_PIPE_MODE_SELECT								
		<p style="text-align: center;">Programming Notes</p> <p>Software must program different Status/Error Buffer addresses between pictures; otherwise the hardware might overwrite previously written data.</p>						
4	31:20	Reserved						
	19	Reserved						
	18	<p>Bitstream Decoder Error Handling Mode</p> <p style="text-align: center;">Programming Notes</p> <p>This modifies how bitstream decoder handles error when there are extra bitstream within the tiles</p> <p>"0"--Decoder will trust the tile byte length and flush extra unused bitstream</p> <p>"1"--Decoder will trust its own decoding logic (ignore byte length). Decoder will start decoding the next tile right after the previous tile finished decoding (bitstream handling)</p>						
	17:16	Reserved						
	15:8	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ						
	7	Reserved						
	6	Reserved						
	5	Reserved						
	4	Reserved						
	3:2	Reserved						
	1	Reserved						
	0	<p>VP9 Flush Handling</p> <p>This modifies how flush is handled</p> <p>"0"--In VP9 mode, all unit done is accounted for before flush;</p> <p>"1"--In VP9 mode, only frame done is accounted for before flush;</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Flush Using Unit Done [Default]</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Flush Using Frame Done</td> </tr> </tbody> </table>	Value	Name	0	Flush Using Unit Done [Default]	1	Flush Using Frame Done
	Value	Name						
0	Flush Using Unit Done [Default]							
1	Flush Using Frame Done							
4.5	31:0	Reserved						
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ							
5	31:0	Reserved						
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ							



HCP_QM_STATE

HCP_QM_STATE									
Source:	VideoCS								
Length Bias:	2								
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HCP_QM_STATE command loads the custom HEVC quantization tables into local RAM and may be issued up to 20 times: 3x Colour Component plus 2x intra/inter plus 4x SizeID minus 4 for the 32x32 chroma components.</p> <p>When the scaling_list_enable_flag is set to disable, the scaling matrix is still sent to the decoder, and with all entries programmed to the same value = 16.</p> <p>This is a picture level state command and is issued in both encoding and decoding processes.</p> <p>Dwords 2-17 form a table for the DCT coefficients, 4 8-bit coefficients/DWord.</p> <ul style="list-style-type: none"> • Size 4x4 for SizeID0, DWords 2-5. • Size 8x8 for SizeID1/2/3, DWords 2-17. 									
SizeID 0 (Table 4-10)									
	4x4	[31:24]	[23:16]	[15:8]	[7:0]				
DWord 2	AC(0,3)	AC(0,2)	AC(0,1)	DC					
DWord 3	AC(1,3)	AC(1,2)	AC(1,1)	AC(1,0)					
DWord 4	AC(2,3)	AC(2,2)	AC(2,1)	AC(2,0)					
DWord 5	AC(3,3)	AC(3,2)	AC(3,1)	AC(3,0)					
SizeID 1, 2, 3 (Table 4-11)									
	8x8	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
DWord 3,2	AC(0,7)	AC(0,6)	AC(0,5)	AC(0,4)	AC(0,3)	AC(0,2)	AC(0,1)	DC	
DWord 5,4	AC(1,7)	AC(1,6)	AC(1,5)	AC(1,4)	AC(1,3)	AC(1,2)	AC(1,1)	AC(1,0)	
DWord 7,6	AC(2,7)	AC(2,6)	AC(2,5)	AC(2,4)	AC(2,3)	AC(2,2)	AC(2,1)	AC(2,0)	
DWord 9,8	AC(3,7)	AC(3,6)	AC(3,5)	AC(3,4)	AC(3,3)	AC(3,2)	AC(3,1)	AC(3,0)	
DWord 11,10	AC(4,7)	AC(4,6)	AC(4,5)	AC(4,4)	AC(4,3)	AC(4,2)	AC(4,1)	AC(4,0)	
DWord 13,12	AC(5,7)	AC(5,6)	AC(5,5)	AC(5,4)	AC(5,3)	AC(5,2)	AC(5,1)	AC(5,0)	
DWord 15,14	AC(6,7)	AC(6,6)	AC(6,5)	AC(6,4)	AC(6,3)	AC(6,2)	AC(6,1)	AC(6,0)	
DWord 17,16	AC(7,7)	AC(7,6)	AC(7,5)	AC(7,4)	AC(7,3)	AC(7,2)	AC(7,1)	AC(7,0)	
DWord	Bit	Description							
0	31:29	Command Type							
Default Value:					3h PARALLEL_VIDEO_PIPE				
Format:					OpCode				



HCP_QM_STATE													
	28:27	Pipeline Type <table border="1"> <tr> <td>Default Value:</td> <td>2h</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	2h	Format:	OpCode							
	Default Value:	2h											
	Format:	OpCode											
	26:23	Media Instruction Opcode <table border="1"> <tr> <td>Default Value:</td> <td>7h Codec/Engine Name</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table> Codec/Engine Name = HCP = 7h	Default Value:	7h Codec/Engine Name	Format:	OpCode							
	Default Value:	7h Codec/Engine Name											
Format:	OpCode												
22:16	Media Instruction Command <table border="1"> <tr> <td>Default Value:</td> <td>4h HCP_QM_STATE</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	4h HCP_QM_STATE	Format:	OpCode								
Default Value:	4h HCP_QM_STATE												
Format:	OpCode												
15:12	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
11:0	Dword Length <table border="1"> <tr> <td>Format:</td> <td>=n</td> </tr> </table> (Excludes Dwords 0, 1). <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>10h</td> <td></td> </tr> </tbody> </table>	Format:	=n	Value	Name	10h							
Format:	=n												
Value	Name												
10h													
1	30:13	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ											
	12:5	DC Coefficient <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Specifies the 8-bit DC coefficient for SizeID 2 and 3. <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">The DC Coefficient must be set to zero for SizeID 0 and 1.</td> </tr> <tr> <td colspan="2">The DC Coefficient must be set to scaling_list_dc_coef_minus8 + 8 for SizeID 2 and 3.</td> </tr> </tbody> </table>	Format:	U8	Programming Notes		The DC Coefficient must be set to zero for SizeID 0 and 1.		The DC Coefficient must be set to scaling_list_dc_coef_minus8 + 8 for SizeID 2 and 3.				
Format:	U8												
Programming Notes													
The DC Coefficient must be set to zero for SizeID 0 and 1.													
The DC Coefficient must be set to scaling_list_dc_coef_minus8 + 8 for SizeID 2 and 3.													
4:3	Color Component <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> Encoder: When RDOQ is enabled, scaling list for all 3 color components must be same. So this field is set to always 0. <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Luma</td> </tr> <tr> <td>1</td> <td>Chroma Cb</td> </tr> <tr> <td>2</td> <td>Chroma Cr</td> </tr> <tr> <td>3</td> <td>Reserved</td> </tr> </tbody> </table>	Format:	U2	Value	Name	0	Luma	1	Chroma Cb	2	Chroma Cr	3	Reserved
Format:	U2												
Value	Name												
0	Luma												
1	Chroma Cb												
2	Chroma Cr												
3	Reserved												



HCP_QM_STATE																	
	2:1	SizeID															
		Format: U2															
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>4x4</td> <td></td> </tr> <tr> <td>1</td> <td>8x8</td> <td></td> </tr> <tr> <td>2</td> <td>16x16</td> <td></td> </tr> <tr> <td>3</td> <td>32x32</td> <td>(Illegal Value for Colour Component Chroma Cr and Cb.)</td> </tr> </tbody> </table>	Value	Name	Description	0	4x4		1	8x8		2	16x16		3	32x32	(Illegal Value for Colour Component Chroma Cr and Cb.)
		Value	Name	Description													
		0	4x4														
	1	8x8															
	2	16x16															
	3	32x32	(Illegal Value for Colour Component Chroma Cr and Cb.)														
	0	Prediction Type															
	Format: U1																
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Intra</td> </tr> <tr> <td>1</td> <td>Inter</td> </tr> </tbody> </table>	Value	Name	0	Intra	1	Inter											
Value	Name																
0	Intra																
1	Inter																
2..17	511:0	See Tables 4-10 and 4-11															



HCP_REF_IDX_STATE

HCP_REF_IDX_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>This is a slice level command used in both encoding and decoding processes. For decoder, it is issued with the HCP_BSD_OBJECT command.</p> <p>Unlike AVC, HEVC allows 16 reference idx entries in each of the L0 and L1 list for a progressive picture. Hence, a max total 32 reference idx in both lists together. The same when the picture is a field picture. Regardless the number of reference idx entries, there are only max 8 reference pictures exist at any one time. Multiple reference idx can point to the same reference picture and can optionally pic a top or bottom field, or frame.</p> <p>For P-Slice, this command is issued only once, representing L0 list. For B-Slice, this command can be issued up to two times, one for L0 list and one for L1 list.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
		Default Value:	7h Codec/Engine Name
		Format:	OpCode
		Codec/Engine Name = HCP = 7h	
	22:16	Media Instruction Command	
		Default Value:	12h HCP_REF_IDX_STATE
		Format:	OpCode
	15:12	Reserved	
Format:		MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	
10h			



HCP_REF_IDX_STATE										
1	31:5	Reserved Format: MBZ								
	4:1	num_ref_idx_l[RefPicListNum]_active_minus1 Format: U4 num_ref_idx_l[RefPicListNum]_active_minus1 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0-14]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0-14]					
	Value	Name								
[0-14]										
0	RefPicListNum Format: U1 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%; text-align: center;">Value</th> <th style="width: 70%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Reference Picture List 0</td> </tr> <tr> <td>1</td> <td>Reference Picture List 1</td> </tr> </tbody> </table>	Value	Name	0	Reference Picture List 0	1	Reference Picture List 1			
Value	Name									
0	Reference Picture List 0									
1	Reference Picture List 1									
2..17	31:16	Reserved Format: MBZ								
	15	list_entry_IX[i]: bottom_field_flag Format: U1 Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 40%; text-align: center;">Value</th> <th style="width: 60%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Bottom Field</td> </tr> <tr> <td>1</td> <td>Top Field</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Not supported in encoder mode.</td> </tr> </tbody> </table>	Value	Name	0	Bottom Field	1	Top Field	Programming Notes	Not supported in encoder mode.
	Value	Name								
0	Bottom Field									
1	Top Field									
Programming Notes										
Not supported in encoder mode.										
14	list_entry_IX[i]: field_pic_flag Format: U1 Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 40%; text-align: center;">Value</th> <th style="width: 60%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Video Frame</td> </tr> <tr> <td>1</td> <td>Video Field</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Not supported in encoder mode.</td> </tr> </tbody> </table>	Value	Name	0	Video Frame	1	Video Field	Programming Notes	Not supported in encoder mode.	
Value	Name									
0	Video Frame									
1	Video Field									
Programming Notes										
Not supported in encoder mode.										



HCP_REF_IDX_STATE									
13	<p>list_entry_IX[i]: LongTermReference</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Short term reference</td> </tr> <tr> <td>1</td> <td>Long term reference</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	Short term reference	1	Long term reference
	Format:	U1							
	Value	Name							
	0	Short term reference							
	1	Long term reference							
12	<p>luma_weight_IX_flag[i]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Default weighted prediction for luma</td> </tr> <tr> <td>1</td> <td>Explicit weighted prediction for Luma</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	Default weighted prediction for luma	1	Explicit weighted prediction for Luma
	Format:	U1							
	Value	Name							
	0	Default weighted prediction for luma							
	1	Explicit weighted prediction for Luma							
11	<p>chroma_weight_IX_flag[i]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Default weighted prediction for Chroma</td> </tr> <tr> <td>1</td> <td>Explicit weighted prediction for Chroma</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	Default weighted prediction for Chroma	1	Explicit weighted prediction for Chroma
	Format:	U1							
	Value	Name							
	0	Default weighted prediction for Chroma							
	1	Explicit weighted prediction for Chroma							
10:8	<p>list_entry_IX[i]: Reference Picture Frame ID (RefAddr[0-7])</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U3</td> </tr> </table> <p>Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.</p> <p>The reference picture frame ID identifies the reference picture associated with the base address defined in Reference Picture Address (RefAddr[0-7]) in the HCP_PIPE_BUF_ADDR_STATE command.</p>	Format:	U3						
	Format:	U3							
7:0	<p>list_entry_IX[i]: Reference Picture tb Value</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.</p> <p>clip(-128,127, CurrentPOC - RefPOC), where RefPOC is the POC value of the reference picture. 8-bit signed.</p> <p>See the "Derivation process for temporal luma motion vector prediction" in the HEVC standard.</p>	Format:	U8						
Format:	U8								



HCP_SLICE_STATE

HCP_SLICE_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>This is a slice level command used in both encoding and decoding processes. For decoder, it is issued with the HCP_BSD_OBJECT command.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
		Default Value:	7h Codec/Engine Name
		Format:	OpCode
		Codec/Engine Name = HCP = 7h	
22:16	Media Instruction Command		
	Default Value:	14h HCP_SLICE_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	
	9h		
7h			
1	31:25	Reserved	
		Format:	MBZ
	24:16	SliceStartCtbY or (slice_start_lcu_y encoder)	
Format:	U9		
Specifies the starting row address of the first coding tree block in the current slice.			



HCP_SLICE_STATE		
	15:9	Reserved Format: MBZ
	8:0	SliceStartCtbX or (slice_start_lcu_x encoder) Format: U9 Specifies the starting column address of the first coding tree block in the current slice.
2	31:25	Reserved Format: MBZ
	24:16	NextSliceStartCtbY or (next_slice_start_lcu_y encoder) Format: U9 Specifies the starting row address of the first coding tree block in the next slice. Must be set to zero when the current slice is the last slice of a picture. For the single slice per frame case, the only slice is also the last slice, so this parameter should be set to a number larger than the frame height (at least +1).
	15	Reserved
	14:9	Reserved Format: MBZ
	8:0	NextSliceStartCtbX or (next_slice_start_lcu_x encoder) Format: U9 Specifies the starting column address of the first coding tree block in the next slice. Must be set to zero when the current slice is the last slice of a picture. For the single slice per frame case, the only slice is also the last slice, so this parameter should be set to a number larger than the frame width (at least +1).
3	31:26	Reserved Format: MBZ
	25	Reserved Format: MBZ
	24	Reserved Format: MBZ
	23	Reserved Format: MBZ
	22	Reserved
	21:17	slice_cr_qp_offset Format: S4 For deblocking purpose, the pic and slice level cr qp offset must be provided separately.



HCP_SLICE_STATE	
PAK needs to perform $\text{final_chroma_cr_qp_offset} = \text{pic_cr_qp_offset} + \text{slice_cr_qp_offset}$.	
Value	Name
14h	-12
15h	-11
16h	-10
17h	-9
18h	-8
19h	-7
1Ah	-6
1Bh	-5
1Ch	-4
1Dh	-3
1Eh	-2
1Fh	-1
0h	0
1h	1
2h	2
3h	3
4h	4
5h	5
6h	6
7h	7
8h	8
9h	9
0Ah	10
0Bh	11
0Ch	12
Programming Notes	
The valid value is from -12 to 12 (or 14h to 0Ch).	
16:12	slice_cb_qp_offset
	Format: S4
	For deblocking purpose, the pic and slice level cb qp offset must be provided separately. PAK needs to perform $\text{final_chroma_cb_qp_offset} = \text{pic_cb_qp_offset} + \text{slice_cb_qp_offset}$.



HCP_SLICE_STATE

Value	Name
14h	-12
15h	-11
16h	-10
17h	-9
18h	-8
19h	-7
1Ah	-6
1Bh	-5
1Ch	-4
1Dh	-3
1Eh	-2
1Fh	-1
0h	0
1h	1
2h	2
3h	3
4h	4
5h	5
6h	6
7h	7
8h	8
9h	9
0Ah	10
0Bh	11
0Ch	12

Programming Notes

The valid value is from -12 to 12 (or 14h to 0Ch).

11:6

SliceQp

Format:	U6
---------	----

Specifies the initial absolute value of QPy quantization parameter for the slice as defined in the Slice Header Semantics section of the HEVC standard.

This signifies only the magnitude of SliceQp. In 8 bit, SliceQp only goes from 0 to 51. But in 10 bit, it needs to go from -12 to 51. There is a sign bit specifies at bit [3] below.



HCP_SLICE_STATE														
	5	<p>slice_temporal_mvp_enable_flag</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U1</td> </tr> </table> <p style="text-align: center;">Programming Notes</p> <p>Must be same for all the slices within a frame in encoder mode (follow spec)</p>	Format:	U1										
	Format:	U1												
	4	<p>dependent_slice_flag</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U1</td> </tr> </table> <p>Decoder only.</p>	Format:	U1										
	Format:	U1												
3	<p>SliceQp Sign Flag</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U1</td> </tr> </table> <p>This specifies the sign bit of SliceQp. This is added for HEVC 10 bit. For 8 bit, SliceQp goes from 0 to 51 so this bit should be zero. In 10 bit, SliceQp goes from -12 to 51 and this bit can be set for negative value.</p>	Format:	U1											
Format:	U1													
2	<p>LastSliceofPic</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U1</td> </tr> </table> <p>This indicates the current slice is the very last slice of the current picture</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Not the last slice of the picture</td> </tr> <tr> <td>1</td> <td>Last slice of the picture</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	Not the last slice of the picture	1	Last slice of the picture					
Format:	U1													
Value	Name													
0	Not the last slice of the picture													
1	Last slice of the picture													
	1:0	<p>slice_type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U2</td> </tr> </table> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>B-slice</td> </tr> <tr> <td>1</td> <td>P-slice</td> </tr> <tr> <td>2</td> <td>I-slice</td> </tr> <tr> <td>3</td> <td>Illegal/Reserved</td> </tr> </tbody> </table>	Format:	U2	Value	Name	0	B-slice	1	P-slice	2	I-slice	3	Illegal/Reserved
	Format:	U2												
	Value	Name												
	0	B-slice												
	1	P-slice												
2	I-slice													
3	Illegal/Reserved													
4	31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
	Format:	MBZ												
	28:26	<p>CollocatedRefIDX</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U3</td> </tr> </table> <p>Collocated Motion Vector Temporal Buffer Index.</p>	Format:	U3										
Format:	U3													
25:23	<p>MaxMergeIDX</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U3</td> </tr> </table> <p>MaxNumMergeCand = 5 - five_minus_max_num_merge_cand - 1.</p>	Format:	U3											
Format:	U3													



HCP_SLICE_STATE			
		Value	Name
		0	0
		1	1
		2	2
		3	3
		4	4
Programming Notes			
The valid value is from 0 to 4 (MaxNumMergeCand = 5 - five_minus_max_num_merge_cand -1)			
22	cabac_init_flag		
	Format:		U1
21:19	luma_log2_weight_denom		
	Format:		U3
18:16	ChromaLog2WeightDenom		
	Format:		U3
15	collocated_from_I0_flag		
	Format:		U1
14	isLowDelay		
	Format:		U1
If the POCs of all pictures in both lists are less than the current POC, then set to one, else set to zero.			
13	mvd_I1_zero_flag		
	Format:		U1
Decoder only.			
12	slice_sao_luma_flag		
	Format:		U1
11	slice_sao_chroma_flag		
	Format:		U1
10	slice_loop_filter_across_slices_enabled_flag		
	Format:		U1
9	Reserved		
	Format:		MBZ
8:5	slice_beta_offset_div2 or (final Beta_Offset_div2 Encoder)		
	Format:		S3
Deblocking filter beta offset. Specified in 2's comp.			



HCP_SLICE_STATE																										
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[1101b,0011b]</td> <td style="text-align: center;">[-3,3]</td> </tr> </tbody> </table>	Value	Name	[1101b,0011b]	[-3,3]																					
Value	Name																									
[1101b,0011b]	[-3,3]																									
	<p>4:1 slice_tc_offset_div2 or (final tc_offset_div2 Encoder)</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">S3</td> </tr> </table> <p>Deblocking filter tc offset. Specified in 2's comp.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[1101b,0011b]</td> <td style="text-align: center;">[-3,3]</td> </tr> </tbody> </table>	Format:	S3	Value	Name	[1101b,0011b]	[-3,3]																			
Format:	S3																									
Value	Name																									
[1101b,0011b]	[-3,3]																									
	<p>0 slice_header_disable_deblocking_filter_flag</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">U1</td> </tr> </table>	Format:	U1																							
Format:	U1																									
5	<p>31:16 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">MBZ</td> </tr> </table>	Format:	MBZ																							
	Format:	MBZ																								
<p>15:0 SliceHeaderLength</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">U16</td> </tr> </table> <p>Decoder only.</p> <p>Specifies the length in bytes of the slice header including the start code. The starting byte of the slice header in the bit stream buffer is indicated by the Indirect Data Start Address in the HCP_BSD_OBJECT command. The ending byte of the slice header in the same bit stream buffer is indicated by the last byte prior to the slice data (CABAC).</p>	Format:	U16																								
Format:	U16																									
6	<p>31:30 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">MBZ</td> </tr> </table>	Format:	MBZ																							
	Format:	MBZ																								
<p>29:26 RoundInter</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">U4</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 30%; text-align: center;">Value</th> <th style="width: 70%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr><td>0h</td><td>+1/32</td></tr> <tr><td>1h</td><td>+2/32</td></tr> <tr><td>2h</td><td>+3/32</td></tr> <tr><td>3h</td><td>+4/32</td></tr> <tr><td>4h</td><td>+5/32 [Default]</td></tr> <tr><td>5h</td><td>+6/32</td></tr> <tr><td>6h</td><td>+7/32</td></tr> <tr><td>7h</td><td>+8/32</td></tr> <tr><td>8h</td><td>+9/32</td></tr> <tr><td>9h</td><td>+10/32</td></tr> <tr><td>Ah</td><td>+11/32</td></tr> </tbody> </table>	Format:	U4	Value	Name	0h	+1/32	1h	+2/32	2h	+3/32	3h	+4/32	4h	+5/32 [Default]	5h	+6/32	6h	+7/32	7h	+8/32	8h	+9/32	9h	+10/32	Ah	+11/32
Format:	U4																									
Value	Name																									
0h	+1/32																									
1h	+2/32																									
2h	+3/32																									
3h	+4/32																									
4h	+5/32 [Default]																									
5h	+6/32																									
6h	+7/32																									
7h	+8/32																									
8h	+9/32																									
9h	+10/32																									
Ah	+11/32																									



HCP_SLICE_STATE																																					
	<table border="1"> <tr><td>Bh</td><td>+12/32</td></tr> <tr><td>Ch</td><td>+13/32</td></tr> <tr><td>Dh</td><td>+14/32</td></tr> <tr><td>Eh</td><td>+15/32</td></tr> <tr><td>Fh</td><td>+16/32</td></tr> </table>	Bh	+12/32	Ch	+13/32	Dh	+14/32	Eh	+15/32	Fh	+16/32																										
Bh	+12/32																																				
Ch	+13/32																																				
Dh	+14/32																																				
Eh	+15/32																																				
Fh	+16/32																																				
	<table border="1"> <tr><td colspan="2">Programming Notes</td></tr> <tr><td colspan="2">Encoder only feature</td></tr> </table>	Programming Notes		Encoder only feature																																	
Programming Notes																																					
Encoder only feature																																					
25:24	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																																		
Format:	MBZ																																				
23:20	RoundIntra <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr><td>0h</td><td>+1/32</td></tr> <tr><td>1h</td><td>+2/32</td></tr> <tr><td>2h</td><td>+3/32</td></tr> <tr><td>3h</td><td>+4/32</td></tr> <tr><td>4h</td><td>+5/32 [Default]</td></tr> <tr><td>5h</td><td>+6/32</td></tr> <tr><td>6h</td><td>+7/32</td></tr> <tr><td>7h</td><td>+8/32</td></tr> <tr><td>8h</td><td>+9/32</td></tr> <tr><td>9h</td><td>+10/32</td></tr> <tr><td>Ah</td><td>+11/32</td></tr> <tr><td>Bh</td><td>+12/32</td></tr> <tr><td>Ch</td><td>+13/32</td></tr> <tr><td>Dh</td><td>+14/32</td></tr> <tr><td>Eh</td><td>+15/32</td></tr> <tr><td>Fh</td><td>+16/32</td></tr> </tbody> </table>	Format:	U4	Value	Name	0h	+1/32	1h	+2/32	2h	+3/32	3h	+4/32	4h	+5/32 [Default]	5h	+6/32	6h	+7/32	7h	+8/32	8h	+9/32	9h	+10/32	Ah	+11/32	Bh	+12/32	Ch	+13/32	Dh	+14/32	Eh	+15/32	Fh	+16/32
Format:	U4																																				
Value	Name																																				
0h	+1/32																																				
1h	+2/32																																				
2h	+3/32																																				
3h	+4/32																																				
4h	+5/32 [Default]																																				
5h	+6/32																																				
6h	+7/32																																				
7h	+8/32																																				
8h	+9/32																																				
9h	+10/32																																				
Ah	+11/32																																				
Bh	+12/32																																				
Ch	+13/32																																				
Dh	+14/32																																				
Eh	+15/32																																				
Fh	+16/32																																				
	<table border="1"> <tr><td colspan="2">Programming Notes</td></tr> <tr><td colspan="2">Encoder only feature</td></tr> </table>	Programming Notes		Encoder only feature																																	
Programming Notes																																					
Encoder only feature																																					
19:0	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																																		
Format:	MBZ																																				



HCP_SLICE_STATE											
7	31:11	Reserved Format: MBZ									
	10	Header Insertion Enable Format: U1 Must be followed by the PAK Insertion Object Command to perform the actual insertion. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>No header insertion into the output bitstream buffer, before the current slice encoded bits.</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>Header insertion into the output bitstream buffer is present, and is before the current slice encoded bits.</td> </tr> </tbody> </table> <div style="text-align: center; background-color: #e6f2ff; padding: 5px; margin-top: 10px;">Programming Notes</div> Must be always enabled. Encoder Only feature	Value	Name	Description	0		No header insertion into the output bitstream buffer, before the current slice encoded bits.	1		Header insertion into the output bitstream buffer is present, and is before the current slice encoded bits.
	Value	Name	Description								
	0		No header insertion into the output bitstream buffer, before the current slice encoded bits.								
1		Header insertion into the output bitstream buffer is present, and is before the current slice encoded bits.									
9	SliceData Enable Format: U1 Must always be enabled. Encoder only feature. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>No operation; no insertion.</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>Slice Data insertion by PAK Object Commands into the output bitstream buffer.</td> </tr> </tbody> </table>	Value	Name	Description	0		No operation; no insertion.	1		Slice Data insertion by PAK Object Commands into the output bitstream buffer.	
Value	Name	Description									
0		No operation; no insertion.									
1		Slice Data insertion by PAK Object Commands into the output bitstream buffer.									
8	Tail Insertion Enable Format: U1 Must be followed by the PAK Insertion Object Command to perform the actual insertion. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>No tail insertion into the output bitstream buffer, after the current slice encoded bits.</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>Tail insertion into the output bitstream buffer is present, and is after the current slice encoded bits. Restriction: Tail insertion is only possible at the end of frame but not in the middle (say slice end)</td> </tr> </tbody> </table> <div style="text-align: center; background-color: #e6f2ff; padding: 5px; margin-top: 10px;">Programming Notes</div> Tail Insertion is allowed only at the end of last slice or last tile of a frame but not in the middle of frame. Also, no multiple tail insertions are allowed. Encoder only feature	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. Restriction: Tail insertion is only possible at the end of frame but not in the middle (say slice end)	
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. Restriction: Tail insertion is only possible at the end of frame but not in the middle (say slice end)									
	7:3	Reserved Format: MBZ									



		HCP_SLICE_STATE	
	2	EmulationByteSliceInsertEnable	
		Format: U1	
		To have PAK outputting SODB or EBSP to the output bitstream buffer.	
		Value	Name
		0	outputting RBSP
	1	outputting EBSP	
	Programming Notes		
	Encoder Only feature		
	1	CabacZeroWordInsertionEnable	
		Format: U1	
To pad the end of a SliceLayer RBSP to meet the encoded size requirement.			
Value		Name	Description
0			No Cabac_Zero_Word Insertion.
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).	
Programming Notes			
Encoder Only feature			
0	Reserved		
	Format: MBZ		
8	31:29	Reserved	
		Format: MBZ	
	28:6	Indirect PAK-BSE Data Start Offset (Write)	
Format: U23			
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 HCP PAK-BSE Object Base Address.			
It is a cacheline-aligned address for the HEVC bitstream data.			
For Write, there is no need to have a data length field. It is assumed the global memory upper 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.			
Value	Name		
0-512MB			



HCP_SLICE_STATE				
		<p style="text-align: center;">Programming Notes</p> <p>Must be zero.</p> <p>Encoder Only feature</p>		
	5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
9	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
15:0	<p>Transformskip_lambda Lambda value used in transform skip calculation</p> <p style="text-align: center;">Programming Notes</p> <p>Encoder Only feature</p>			
10	31:24	<p>Transformskip_numnonzerocoeffs_factor1 Multiplying factor with number of non-zero coefficients for non-Skip pipe calculation</p> <p style="text-align: center;">Programming Notes</p> <p>Encoder Only feature</p>		
	23:16	<p>Transformskip_numzerocoeffs_factor1 Multiplying factor with number of zero coefficients for non-Skip pipe calculation</p> <p style="text-align: center;">Programming Notes</p> <p>Encoder Only feature</p>		
	15:8	<p>Transformskip_numnonzerocoeffs_factor0 Multiplying factor with number of non-zero coefficients for skip pipe calculation</p> <p style="text-align: center;">Programming Notes</p> <p>Encoder Only</p>		
	7:0	<p>Transformskip_numzerocoeffs_factor0 Multiplying factor with number of zero coefficients for skip pipe calculation</p> <p style="text-align: center;">Programming Notes</p> <p>Encoder Only feature</p>		



HCP_SURFACE_STATE

HCP_SURFACE_STATE				
Source:	VideoCS			
Length Bias:	2			
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HCP_SURFACE_STATE command is responsible for defining the frame buffer pitch and the offset of the chroma component.</p> <p>This is a picture level state command and is shared by both encoding and decoding processes.</p> <p>Note : Only NV12 and Tile Y are being supported for HEVC. Hence full pitch and interleaved UV is always in use. U and V Xoffset must be set to 0; U and V Yoffset must be 16-pixel aligned. This Surface State is not the same as that of the 3D engine and of the MFX pipeline.</p>				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	3h PARALLEL_VIDEO_PIPE	
		Format:	OpCode	
	28:27	28:27	Pipeline Type	
			Default Value:	2h
			Format:	OpCode
	26:23	26:23	Media Instruction Opcode	
			Default Value:	7h Codec/Engine Name
			Format:	OpCode
			Codec/Engine Name = HCP = 7h	
22:16	22:16	Media Instruction Command		
		Default Value:	1h HCP_SURFACE_STATE	
		Format:	OpCode	
15:12	15:12	Reserved		
		Format:	MBZ	
11:0	11:0	Dword Length		
		Format:	=n	
		(Excludes Dwords 0, 1).		
		Value	Name	
1h				
1	31:28	Surface Id		
		Format:	U4	



HCP_SURFACE_STATE			
	Value	Name	Description
	0h	HEVC: For current decoded Picture	8-bit uncompressed data
	1h	Source Input Picture (encoder)	8-bit uncompressed data
	2h	Prev Reference Picture	(VP9 only) Previous Reference
	3h	Golden Reference Picture	(VP9 only) Golden Reference
	4h	AltRef Reference Picture	(VP9 only) AltRef Reference
27:17	Reserved		
	Format:	MBZ	
16:0	Surface Pitch Minus1		
	Format:	U17-1	
	This field specifies the surface pitch in (#Bytes - 1).		
	Programming Notes		
	<ul style="list-style-type: none"> For tiled surfaces, the pitch must be a multiple of the tile width. If Half Pitch for Chroma is set, this field must be a multiple of two tile widths for tiled surfaces, or a multiple of 2 bytes for linear surfaces. 		
	For Y-tiled surfaces: Range = [127, 524287]->[128B,256KB] = [1 tile, 2048 tiles].		
2	31:28	Surface Format	
	Format:	U4	
	Specifies the format of the surface.		
	Value	Name	Programming Notes
	0h-2h	Reserved	
	3h	P010Variant	P010Variant is a modified P010 format, >8 bit planar 420 with MSB together and LSB at an offset in x direction where the x-offset should be 32-bit aligned.
	4h	PLANAR_420_8	
	5h-Ch	Reserved	
	Dh	P010	
	Eh	P016	This is added for VP9 8./10/12 bit decode
	Fh	Reserved	
27:15	Reserved		
	Format:	MBZ	
14:0	Y Offset for U(Cb) in pixel		
	Format:	U15_Pixel_Row_Offset	
	This field specifies the vertical offset in rows 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.		



HCP_SURFACE_STATE		
		Programming Notes
		<p>The following restrictions are applicable when Memory compression is enabled.</p> <ul style="list-style-type: none"> • For PLANAR_420 surface formats, the alignment of this field follows the tile mode described in bits 14:13 of the Memory Address Attributes table. • TileY (legacy 4k) - 8 pixel aligned • TileYF (New 4k) - 64 pixel aligned • TileYS (64k) - 256 pixel aligned
		When Memory compression is not enabled, This field should be multiple of 8 pixels.



HCP_TILE_STATE

HCP_TILE_STATE							
Source:	VideoCS						
Length Bias:	2						
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p>							
DWord	Bit	Description					
0	31:29	Command Type <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>3h PARALLEL_VIDEO_PIPE</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	3h PARALLEL_VIDEO_PIPE	Format:	OpCode	
		Default Value:	3h PARALLEL_VIDEO_PIPE				
		Format:	OpCode				
	28:27	Pipeline Type <table border="1" style="width: 100%;"> <tr> <td style="width: 75%;">Default Value:</td> <td>2h</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	2h	Format:	OpCode	
		Default Value:	2h				
	Format:	OpCode					
26:23	Media Instruction Opcode <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>7h Codec/Engine Name</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table> Codec/Engine Name = HCP = 7h	Default Value:	7h Codec/Engine Name	Format:	OpCode		
	Default Value:	7h Codec/Engine Name					
	Format:	OpCode					
22:16	Media Instruction Command <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>11h HCP_TILE_STATE</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	11h HCP_TILE_STATE	Format:	OpCode		
	Default Value:	11h HCP_TILE_STATE					
Format:	OpCode						
15:12	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 75%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
11:0	Dword Length <table border="1" style="width: 100%;"> <tr> <td style="width: 75%;">Format:</td> <td>=n</td> </tr> </table> (Excludes Dwords 0, 1). <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Bh</td> <td></td> </tr> </tbody> </table>	Format:	=n	Value	Name	Bh	
Format:	=n						
Value	Name						
Bh							
1	31:10	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 75%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
		Format:	MBZ				
	NumTileColumnsMinus1 <table border="1" style="width: 100%;"> <tr> <td style="width: 75%;">Format:</td> <td>U5</td> </tr> </table> Specifies the number of tile columns in Ctbs per picture.	Format:	U5				
Format:	U5						
4:0	NumTileRowsMinus1 <table border="1" style="width: 100%;"> <tr> <td style="width: 75%;">Format:</td> <td>U5</td> </tr> </table> Specifies the number of tile rows in Ctbs per picture.	Format:	U5				
Format:	U5						



HCP_TILE_STATE		
2	31:24	Ctb column position of tile column 3 Format: <input type="text"/> U8
	23:16	Ctb column position of tile column 2 Format: <input type="text"/> U8
	15:8	Ctb column position of tile column 1 Format: <input type="text"/> U8
	7:0	Ctb column position of tile column 0 Format: <input type="text"/> U8
3	31:24	Ctb column position of tile column 7 Format: <input type="text"/> U8
	23:16	Ctb column position of tile column 6 Format: <input type="text"/> U8
	15:8	Ctb column position of tile column 5 Format: <input type="text"/> U8
	7:0	Ctb column position of tile column 4 Format: <input type="text"/> U8
4	31:24	Ctb column position of tile column 11 Format: <input type="text"/> U8
	23:16	Ctb column position of tile column 10 Format: <input type="text"/> U8
	15:8	Ctb column position of tile column 9 Format: <input type="text"/> U8
	7:0	Ctb column position of tile column 8 Format: <input type="text"/> U8
5	31:24	Ctb column position of tile column 15 Format: <input type="text"/> U8
	23:16	Ctb column position of tile column 14 Format: <input type="text"/> U8
	15:8	Ctb column position of tile column 13 Format: <input type="text"/> U8
	7:0	Ctb column position of tile column 12 Format: <input type="text"/> U8
6	31:24	Ctb column position of tile column 19 Format: <input type="text"/> U8
	23:16	Ctb column position of tile column 18



HCP_TILE_STATE		
		Format: U8
	15:8	Ctb column position of tile column 17 Format: U8
	7:0	Ctb column position of tile column 16 Format: U8
7	31:24	Ctb row position of tile row 3 Format: U8
	23:16	Ctb row position of tile row 2 Format: U8
	15:8	Ctb row position of tile row 1 Format: U8
	7:0	Ctb row position of tile row 0 Format: U8
8	31:24	Ctb row position of tile row 7 Format: U8
	23:16	Ctb row position of tile row 6 Format: U8
	15:8	Ctb row position of tile row 5 Format: U8
	7:0	Ctb row position of tile row 4 Format: U8
9	31:24	Ctb row position of tile row 11 Format: U8
	23:16	Ctb row position of tile row 10 Format: U8
	15:8	Ctb row position of tile row 9 Format: U8
	7:0	Ctb row position of tile row 8 Format: U8
10	31:24	Ctb row position of tile row 15 Format: U8
	23:16	Ctb row position of tile row 14 Format: U8
	15:8	Ctb row position of tile row 13 Format: U8



HCP_TILE_STATE		
	7:0	Ctb row position of tile row 12 Format: U8
11	31:24	Ctb row position of tile row 19 Format: U8
	23:16	Ctb row position of tile row 18 Format: U8
	15:8	Ctb row position of tile row 17 Format: U8
	7:0	Ctb row position of tile row 16 Format: U8
12	31:16	Reserved Format: MBZ
	15:8	Ctb row position of tile row 21 Format: U8
	7:0	Ctb row position of tile row 20 Format: U8



HCP_VP9_PAK_OBJECT

HCP_VP9_PAK_OBJECT			
Source:		VideoCS	
Length Bias:		2	
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Reserved	
	22:16	Media Instruction Command	
		Default Value:	35h HCP_VP9_PAK_OBJECT
		Format:	OpCode
15:12	Reserved		
	Format:	MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	
	0h		
1	31	IsLastSB Flag (of Tile)	
		Format:	Enable
		Value	Name
		0	False
	1	True	
	30	Reserved	
		Format:	MBZ
	29:24	CU_count_minus1	
		Format:	U6
	Number of CUs in the current SB = CU_count_minus1 + 1. Minimum, there must be 1 CU in a SB.		
23:21	Reserved		
	Format:	MBZ	



HCP_VP9_PAK_OBJECT		
	20:0	split_coding_unit_flag[x0][y0]
		Format: U21
		Bit 20 Split_flag_level0
		Bit 19:16 Split_flag_level1 [19:16] is in raster order. Bit16 is for partition0 in raster order.
		Bit 15:12 Split_flag_level2_level1part3 Split flags for bit19 partition. [15:12] is in raster order. Bit12 is for partition0 in raster order.
		Bit 11:8 Split_flag_level2_level1part2 Split flags for bit18 partition. [11:8] is in raster order. Bit8 is for partition0 in raster order.
		Bit 7:4 Split_flag_level2_level1part1 Split flags for bit17 partition. [7:4] is in raster order. Bit4 is for partition0 in raster order.
Bit 3:0 Split_flag_level2_level1part0 Split flags for bit16 partition. [3:0] is in raster order. Bit0 is for partition0 in raster order.		
2	31:16	Current SB Y Addr)
	Format: U16	
	15:0	Current SB X Addr
	Format: U16	



HCP_VP9_PIC_STATE

HCP_VP9_PIC_STATE			
Source:		VideoCS	
Length Bias:		2	
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
		Default Value:	7h Codec/Engine Name
		Format:	OpCode
		Codec/Engine Name = HUC = Bh	
22:16	Media Instruction Command		
	Default Value:	30h HCP_VP9_PIC_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	Programming Notes
	Bh	Decoder DW Length	Only Up to DW12 should be programmed for decoder
1Eh	Encoder DW Length	All DWs should be programmed for encoder	
1	31:30	Reserved	
		Format:	MBZ
	29:16	Frame Height In Pixels Minus 1	
Format:		U14	
<p>Specifies the height of the decoded picture in units of 8 pixels, which is the minimum coding block size. The decoded picture height in units of luma samples equals</p> $(\text{FrameHeightInMinBlocksMinus1} + 1) * 8 - 1$ <p><i>For Encoder Partial SB:</i> <i>Kernel, on last SB (at picture edges), splits the SB into as small as possible CUs to have picture boundaries aligned to CU boundary.</i></p>			



HCP_VP9_PIC_STATE															
	<p><i>Kernel does not count/include CUs outside of the picture in PakObj/CU_record respectively. Driver sets up a SB aligned (both in X/Y direction) surface.</i></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[4096,16383]</td> <td style="text-align: center;">4K_TO_16K</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Decoder supports 16K image and 8K video. Encoder only supports upto 8K.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0-6</td> <td>Invalid (multiple of 8 pixels)</td> </tr> <tr> <td style="text-align: center;">7-8191</td> <td>8-8K Pixels (Decoder and Encoder)</td> </tr> <tr> <td style="text-align: center;">8192-16383</td> <td>8K-16K Pixels (Decoder only)</td> </tr> </tbody> </table>	Value	Name	[4096,16383]	4K_TO_16K	Value	Description	0-6	Invalid (multiple of 8 pixels)	7-8191	8-8K Pixels (Decoder and Encoder)	8192-16383	8K-16K Pixels (Decoder only)		
Value	Name														
[4096,16383]	4K_TO_16K														
Value	Description														
0-6	Invalid (multiple of 8 pixels)														
7-8191	8-8K Pixels (Decoder and Encoder)														
8192-16383	8K-16K Pixels (Decoder only)														
15:14	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ														
13:0	<p>Frame Width In Pixels Minus 1</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">U14</td> </tr> </table> <p>Specifies the width of the decoded picture in units of minimum coding block size. The decoded picture width in units of luma samples equals (FrameWidthInMinBlocksMinus1+ 1) *8 – 1 This should be programmed to a multiple of 8 pixels minus 1. <i>For Encoder Partial SB: Kernel, on last SB (at picture edges), splits the SB into as small as possible CUs to have picture boundaries aligned to CU boundary. Kernel does not count/include CUs outside of the picture in PakObj/CU_record respectively. Driver sets up a SB aligned (both in X/Y direction) surface.</i></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[4096,16383]</td> <td style="text-align: center;">4K_TO_16K</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Decoder supports 16K image and 8K video. Encoder only supports upto 8K.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0-6</td> <td>Invalid (multiple of 8 pixels)</td> </tr> <tr> <td style="text-align: center;">7-8191</td> <td>8-8K Pixels (Decoder and Encoder)</td> </tr> <tr> <td style="text-align: center;">8192-16383</td> <td>8K-16K Pixels (Decoder only)</td> </tr> </tbody> </table>	Format:	U14	Value	Name	[4096,16383]	4K_TO_16K	Value	Description	0-6	Invalid (multiple of 8 pixels)	7-8191	8-8K Pixels (Decoder and Encoder)	8192-16383	8K-16K Pixels (Decoder only)
Format:	U14														
Value	Name														
[4096,16383]	4K_TO_16K														
Value	Description														
0-6	Invalid (multiple of 8 pixels)														
7-8191	8-8K Pixels (Decoder and Encoder)														
8192-16383	8K-16K Pixels (Decoder only)														
2	<p style="text-align: center;">31</p> <p>Segment ID StreamIn Enable</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">Enable</td> </tr> </table> <p>Indicates SegmentID from previous frame needs to be streamIn for Segment ID prediction</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">Enable</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	0	Disable	1	Enable						
Format:	Enable														
Value	Name														
0	Disable														
1	Enable														



HCP_VP9_PIC_STATE													
	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Deocder Only: SegmentIDStreamInEnable = error_resilient_mode OR intra_only OR VP9_KEY_FRAME</td> </tr> </table>	Programming Notes		Deocder Only: SegmentIDStreamInEnable = error_resilient_mode OR intra_only OR VP9_KEY_FRAME									
Programming Notes													
Deocder Only: SegmentIDStreamInEnable = error_resilient_mode OR intra_only OR VP9_KEY_FRAME													
30	<p>Segment ID StreamOut Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Indicates SegmentID of current frame needs to be streamOut for next frame</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 40%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> </tr> <tr> <td>1</td> <td>Enable</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Deocder Only: SegmentIDStreamOutEnable = segementation_enabled AND segmentation_update_map</td> </tr> </table>	Format:	Enable	Value	Name	0	Disable	1	Enable	Programming Notes		Deocder Only: SegmentIDStreamOutEnable = segementation_enabled AND segmentation_update_map	
Format:	Enable												
Value	Name												
0	Disable												
1	Enable												
Programming Notes													
Deocder Only: SegmentIDStreamOutEnable = segementation_enabled AND segmentation_update_map													
29	<p>Lossless Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bitSet to indicate lossless coding mode.</p> <p>In encoder mode, software has to set tx_mode to 4x4only and all tu_size's in CU record as 4x4 for entire frame. Software also has to program such that final_qindex=0 and final_filter_level=0 following the Quant Scale and Filter Level Table in Segmentation State section. Hardware forces Hadamard Tx when this bit is set. When Lossless Mode is on, BRC has to be off.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 40%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Normal Mode</td> </tr> <tr> <td>1</td> <td>Loless Mode</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	0	Normal Mode	1	Loless Mode				
Format:	Enable												
Value	Name												
0	Normal Mode												
1	Loless Mode												
28	<p>Segmentation Temporal Update</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Indicates whether segID is decoding from bitstream or predicted from previous frame.</p> <p>In encoder Mode it should use either from previous frame or streamIn</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Decode segID from bitstream</td> </tr> <tr> <td>1h</td> <td>Get segID either from bitstream or from previous frame</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Decoder Only: For KEY_FRAME or INTRA_ONLY frame, this bit should be set to "0". Note: Driver should override this flag to "0" in KEY_FRAME or INTRA_ONLY frame even if this</td> </tr> </table>	Format:	Enable	Value	Name	0h	Decode segID from bitstream	1h	Get segID either from bitstream or from previous frame	Programming Notes		Decoder Only: For KEY_FRAME or INTRA_ONLY frame, this bit should be set to "0". Note: Driver should override this flag to "0" in KEY_FRAME or INTRA_ONLY frame even if this	
Format:	Enable												
Value	Name												
0h	Decode segID from bitstream												
1h	Get segID either from bitstream or from previous frame												
Programming Notes													
Decoder Only: For KEY_FRAME or INTRA_ONLY frame, this bit should be set to "0". Note: Driver should override this flag to "0" in KEY_FRAME or INTRA_ONLY frame even if this													



HCP_VP9_PIC_STATE												
	bit decoded from bitstream is different. This is for hardware optimization. This override does not affect bitstream decoding other than uncompressed header.											
27	<p>Segmentation Update Map</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Indicates how hardware determines segmentation ID</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Intra block: segment ID is zero Inter block: get segment ID from previous frame (streamIN)</td> </tr> <tr> <td>1h</td> <td></td> <td>Intra block: decode segment ID from bitstream. Inter block: determines from segmentation_temporal_update setting</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0h		Intra block: segment ID is zero Inter block: get segment ID from previous frame (streamIN)	1h		Intra block: decode segment ID from bitstream. Inter block: determines from segmentation_temporal_update setting
Format:	Enable											
Value	Name	Description										
0h		Intra block: segment ID is zero Inter block: get segment ID from previous frame (streamIN)										
1h		Intra block: decode segment ID from bitstream. Inter block: determines from segmentation_temporal_update setting										
26	<p>Segmentation Enabled</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Indicate if segmentation is enabled or not</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>All blocks are implied to belong to segment 0</td> </tr> <tr> <td>1h</td> <td>SegID determination depends on segmentation_update_map setting</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	0h	All blocks are implied to belong to segment 0	1h	SegID determination depends on segmentation_update_map setting			
Format:	Enable											
Value	Name											
0h	All blocks are implied to belong to segment 0											
1h	SegID determination depends on segmentation_update_map setting											
25:23	<p>Sharpness Level</p> <table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Specify the sharpness level, as one of regular deblocking strength control.</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">Set to 0 to disable the use of sharpness control</td> </tr> </table>	Format:	U3	Programming Notes		Set to 0 to disable the use of sharpness control						
Format:	U3											
Programming Notes												
Set to 0 to disable the use of sharpness control												
22:17	<p>Filter Level</p> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Specify the Filter level, as one of deblocking strength control</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">Set to 0 to disable the use of level control</td> </tr> </table>	Format:	U6	Programming Notes		Set to 0 to disable the use of level control						
Format:	U6											
Programming Notes												
Set to 0 to disable the use of level control												
16	<p>Frame Parallel Decoding Mode</p> <p>Indicates if parallel decoding mode is enabled. This bit should come from Uncompressed header. Together with Error Resilient mode, they decide the value of AdaptProbabilityFlag.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> </tr> <tr> <td>1</td> <td>Enable</td> </tr> </tbody> </table>	Value	Name	0	Disable	1	Enable					
Value	Name											
0	Disable											
1	Enable											
15	<p>Error Resilient Mode</p> <p>Indicates if error resilient mode is enabled. This bit should come from Uncompressed header. When error resilient is 1, Frame Parallel Decoding Mode will be 1, and Refresh Frame Context will be 0. When error resilient is 0, Frame Parallel Decoding Mode and Refresh Frame Context read from bit stream. Together with Frame Parallel Decoding mode, they decide the value of AdaptProbabilityFlag.</p>											



HCP_VP9_PIC_STATE													
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Enable</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> </td> </tr> </tbody> </table>	Value	Name	0	Disable	1	Enable	Programming Notes					
Value	Name												
0	Disable												
1	Enable												
Programming Notes													
14	<p>Refresh Frame Context Indicates if Frame Context should be refresh. This bit should come from Uncompressed header</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Enable</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">Decoder Only</td> </tr> </tbody> </table>	Value	Name	0	Disable	1	Enable	Programming Notes		Decoder Only			
Value	Name												
0	Disable												
1	Enable												
Programming Notes													
Decoder Only													
13	<p>Last Frame Type It indicates the frame type of previous frame (Key or Non-Key Frame)</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Key Frame</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Non Key Frame</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">Not used in Encoder Mode</td> </tr> </tbody> </table>	Value	Name	0	Key Frame	1	Non Key Frame	Programming Notes		Not used in Encoder Mode			
Value	Name												
0	Key Frame												
1	Non Key Frame												
Programming Notes													
Not used in Encoder Mode													
12	<p>Selectable TX Mode</p> <table border="1"> <tr> <td>Format:</td> <td style="text-align: center;">U1</td> </tr> </table> <p>Indicates if tx_mode is selectable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Encoder does not pack tu_size into bitstream. This helps reduce bitstream size further.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Encoder packs tu_size into bitstream.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">HW always picks tu_size from CU record of pak_obj. SW responsibility to set tu_size correct.</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	Encoder does not pack tu_size into bitstream. This helps reduce bitstream size further.	1	Encoder packs tu_size into bitstream.	Programming Notes		HW always picks tu_size from CU record of pak_obj. SW responsibility to set tu_size correct.	
Format:	U1												
Value	Name												
0	Encoder does not pack tu_size into bitstream. This helps reduce bitstream size further.												
1	Encoder packs tu_size into bitstream.												
Programming Notes													
HW always picks tu_size from CU record of pak_obj. SW responsibility to set tu_size correct.													
11	<p>Hybrid Prediction Mode</p> <table border="1"> <tr> <td>Format:</td> <td style="text-align: center;">U1</td> </tr> </table> <p>Indicates if comp_pred_mode is hybrid</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>comp_prediction_mode!=HYBRID, Encoder does not pack comp_pred_mode [interpreted_comp in pak_obj] into bitstream.</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	comp_prediction_mode!=HYBRID, Encoder does not pack comp_pred_mode [interpreted_comp in pak_obj] into bitstream.						
Format:	U1												
Value	Name												
0	comp_prediction_mode!=HYBRID, Encoder does not pack comp_pred_mode [interpreted_comp in pak_obj] into bitstream.												



HCP_VP9_PIC_STATE															
1	comp_prediction_mode==HYBRID, Encoder packs comp_pred_mode into bitstream. This helps reduce bitstream size further.														
10	<p>Use Prev in Find MV References</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>0: Temporal MV buffer is not available for MV prediction 1: Temporal MV buffer is available for MV prediction This is set to 0 when:</p> <ul style="list-style-type: none"> • The last picture has a different size • Current picture is error-resilient mode • Current picture is intra_only, or keyframe • Last picture was intra_only or keyframe • Last picture was not a displayed picture. 	Format:	Enable												
Format:	Enable														
9:7	<p>Ref Frame Sign Bias[0..2]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U3</td> </tr> </table> <p>Reference Frame sign bias (not including intra reference) Bit[7] – Sign Bias of Last Frame Bit[8] – Sign Bias of Golden Frame Bit[9] – Sign Bias of AltRef Frame</p>	Format:	U3												
Format:	U3														
6:4	<p>Mcomp Filter Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U3</td> </tr> </table> <p>Indicate Motion Compensation Filter type. If set to 4, encoder uses modes in pak_obj command.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>Eight-tap</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>Eight-tap-Smooth</td> </tr> <tr> <td style="text-align: center;">2h</td> <td>Eight-tap-Sharp</td> </tr> <tr> <td style="text-align: center;">3h</td> <td>Bilinear</td> </tr> <tr> <td style="text-align: center;">4h</td> <td>Switchable</td> </tr> </tbody> </table>	Format:	U3	Value	Name	0h	Eight-tap	1h	Eight-tap-Smooth	2h	Eight-tap-Sharp	3h	Bilinear	4h	Switchable
Format:	U3														
Value	Name														
0h	Eight-tap														
1h	Eight-tap-Smooth														
2h	Eight-tap-Sharp														
3h	Bilinear														
4h	Switchable														
3	<p>Allow Hi Precision MV</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>Indicate high precision mode for Motion Vector prediction</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>Normal mode</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>High Precision mode</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	0h	Normal mode	1h	High Precision mode						
Format:	Enable														
Value	Name														
0h	Normal mode														
1h	High Precision mode														



HCP_VP9_PIC_STATE														
2	IntraOnly Flag													
	Format: Enable													
	Indicates intra-only for inter pics. MBZ for keyframes.													
Programming Notes														
Used for Non-displayable picture; SW responsibility to make sure no Inter block in pak_obj of this frame														
<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 10%;">Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Inter Frame use both intra/inter-blocks</td> </tr> <tr> <td>1</td> <td>Inter frame use only inta-blocks</td> </tr> </tbody> </table>			Value	Description	0	Inter Frame use both intra/inter-blocks	1	Inter frame use only inta-blocks						
Value	Description													
0	Inter Frame use both intra/inter-blocks													
1	Inter frame use only inta-blocks													
1	Adapt Probabilities Flag													
	Format: Enable													
	Indicates that the probabilities used to decode this frame should be adapted													
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0: Do not adapt (error resilient or frame_parallel_mode are set)</td> </tr> <tr> <td>1h</td> <td>1: Adapt (not error resilient and not frame_parallel_mode)</td> </tr> </tbody> </table>		Value	Name	0h	0: Do not adapt (error resilient or frame_parallel_mode are set)	1h	1: Adapt (not error resilient and not frame_parallel_mode)						
Value	Name													
0h	0: Do not adapt (error resilient or frame_parallel_mode are set)													
1h	1: Adapt (not error resilient and not frame_parallel_mode)													
Programming Notes														
Only forward adaptation is supported in encoder mode.														
0	Frame Type													
	Format: U1													
	Specifies the VP9 frame type													
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Key Frame</td> </tr> <tr> <td>1h</td> <td>Inter Frame</td> </tr> </tbody> </table>		Value	Name	0h	Key Frame	1h	Inter Frame						
Value	Name													
0h	Key Frame													
1h	Inter Frame													
3	31:28	Profile Level												
		Format: U4												
		This indicates VP9 Profile level from bitstream												
		<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Profile_0</td> <td>Profile 0 only supports 8 bit 420 only</td> </tr> <tr> <td>2</td> <td>Profile_2</td> <td>Profile 2 only supports 10 bits 420 only</td> </tr> </tbody> </table>		Value	Name	Description	0	Profile_0	Profile 0 only supports 8 bit 420 only	2	Profile_2	Profile 2 only supports 10 bits 420 only		
		Value	Name	Description										
	0	Profile_0	Profile 0 only supports 8 bit 420 only											
	2	Profile_2	Profile 2 only supports 10 bits 420 only											
	27:24	BitDepthMinus8												
		Format: U4												
This indicates the bitdepth (minus 8) of the pixels														
<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 10%;">Value</th> <th>Name</th> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Bitdepth_8</td> <td>It indicates pixel bitdepth is 8. Only profile 0 is allowed in this mode.</td> </tr> <tr> <td>2</td> <td>Bitdepth_10</td> <td>It indicates pixel bitdepth is 10. Only profile 2 is allowed in this mode.</td> </tr> <tr> <td>4</td> <td>Bitdepth_12</td> <td>It indicates pixel bitdepth is 12. Only profile 2 is allowed in this mode.</td> </tr> </tbody> </table>		Value	Name	Programming Notes	0	Bitdepth_8	It indicates pixel bitdepth is 8. Only profile 0 is allowed in this mode.	2	Bitdepth_10	It indicates pixel bitdepth is 10. Only profile 2 is allowed in this mode.	4	Bitdepth_12	It indicates pixel bitdepth is 12. Only profile 2 is allowed in this mode.	
Value		Name	Programming Notes											
0	Bitdepth_8	It indicates pixel bitdepth is 8. Only profile 0 is allowed in this mode.												
2	Bitdepth_10	It indicates pixel bitdepth is 10. Only profile 2 is allowed in this mode.												
4	Bitdepth_12	It indicates pixel bitdepth is 12. Only profile 2 is allowed in this mode.												



HCP_VP9_PIC_STATE		
23:22	Chroma Sampling Format	
	Format:	U2
	This indicates the chroma sampling format of the bitstream	
	Value	Name Programming Notes
	0	Format_420 Chroma Format 420, supported by profile 0 and 2
Programming Notes		
Currently only 420 is supported (in profile 0 and 2). All other modes are not valid. Value 0: Format 420: Profile 0 and 2 only: Supported Value 1: Format 422: Profile 1 and 3 only: Currently NOT supported Value 2: Format 444: Profile 1 and 3 only: Currently NOT supported		
21	Reserved	
	Format:	MBZ
20:10	Reserved	
	Format:	MBZ
9:8	Log2 Tile Row	
	Format:	U2
	This indicates the number of tile rows (log2).	
	Value	Name
	0h	1 Tile Row
	1h	2 Tile Row
2h	4 Tile Row	
Programming Notes		
In encoder mode, create tiles either in horizontal or vertical direction but not in both directions		
7:4	Reserved	
	Format:	MBZ
3:0	Log2 Tile Column	
	Format:	U4
	This indicates the number of tile rows (log2).	
	Value	Name Programming Notes
	0h	1 Tile Column
	1h	2 Tile Column
2h	4 Tile Column	In encoder mode, create tiles either in horizontal or vertical direction but not in both directions
3h	8 Tile Column	



HCP_VP9_PIC_STATE											
		<table border="1"> <tr> <td>4h</td> <td>16 Tile Column</td> <td></td> </tr> <tr> <td>5h</td> <td>32 Tile Column</td> <td></td> </tr> <tr> <td>6h</td> <td>64 Tile Column</td> <td></td> </tr> </table>	4h	16 Tile Column		5h	32 Tile Column		6h	64 Tile Column	
4h	16 Tile Column										
5h	32 Tile Column										
6h	64 Tile Column										
		Programming Notes In encoder mode, create tiles either in horizontal or vertical direction but not in both directions									
4	31:16	Horizontal Scale Factor for LAST Format: <table border="1"><tr><td></td><td>U2.14</td></tr></table> This indicates the scaling factor between current frame and the last reference frame Set to $(LastWidth * 2^{14})/CurrentWidth$		U2.14							
		U2.14									
15:0	Vertical Scale Factor for LAST Format: <table border="1"><tr><td></td><td>U2.14</td></tr></table> This indicates the scaling factor between current frame and the last reference frame Set to $(LastHeight * 2^{14})/CurrentHeight$		U2.14								
	U2.14										
5	31:16	Horizontal Scale Factor for GOLDEN Format: <table border="1"><tr><td></td><td>U2.14</td></tr></table> This indicates the scaling factor between current frame and the golden reference frame Set to $(LastWidth * 2^{14})/CurrentWidth$		U2.14							
		U2.14									
15:0	Vertical Scale Factor for GOLDEN Format: <table border="1"><tr><td></td><td>U2.14</td></tr></table> This indicates the scaling factor between current frame and the golden reference frame Set to $(LastHeight * 2^{14})/CurrentHeight$		U2.14								
	U2.14										
6	31:16	Horizontal Scale Factor for ALTREF Format: <table border="1"><tr><td></td><td>U2.14</td></tr></table> This indicates the scaling factor between current frame and the altref reference frame Set to $(LastWidth * 2^{14})/CurrentWidth$		U2.14							
		U2.14									
15:0	Vertical Scale Factor for ALTREF Format: <table border="1"><tr><td></td><td>U2.14</td></tr></table> This indicates the scaling factor between current frame and the altref reference frame Set to $(LastHeight * 2^{14})/CurrentHeight$		U2.14								
	U2.14										



HCP_VP9_PIC_STATE		
7	31:30	Reserved Format: MBZ
	29:16	Last Frame Hieght In Pixels Minus 1 Format: U14 Specifies the height of the Last picture in units of pixels. The Last picture height in units of luma samples equals (LastFrameHeightInMinBlocksMinus1+ 1) Value Description 0-16383 1-16K Pixels (encoder only goes to 8K)
	15:14	Reserved Format: MBZ
	13:0	Last Frame Width In Pixels Minus 1 Format: U14 Specifies the width of the Last picture in units of pixels. The Last picture width in units of luma samples equals (LastFrameWidthtInMinBlocksMinus1+ 1) Value Description 0-16383 1-16K Pixels (encoder only goes to 8K)
8	31:30	Reserved Format: MBZ
	29:16	Golden Frame Hieght In Pixels Minus 1 Format: U14 Specifies the height of the Last picture in units of pixels. The Golden picture height in units of luma samples equals (LastFrameHeightInMinBlocksMinus1+ 1) Value Description 0-16383 1-16K Pixels (encoder only goes to 8K)
	15:14	Reserved Format: MBZ
	13:0	Golden Frame Width In Pixels Minus 1 Format: U14 Specifies the width of the Last picture in units of pixels. The Golden picture width in units of luma samples equals (LastFrameWidthtInMinBlocksMinus1+ 1) Value Description 0-16383 1-16K Pixels (encoder only goes to 8K)
9	31:30	Reserved Format: MBZ



HCP_VP9_PIC_STATE							
	<p>29:16 Altref Frame Hieght In Pixels Minus 1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U14</td> </tr> </table> <p>Specifies the height of the Last picture in units of pixels. The Altref picture height in units of luma samples equals (LastFrameHeightInMinBlocksMinus1 + 1)</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-16383</td> <td>1-16K Pixels (encoder only goes to 8K)</td> </tr> </tbody> </table>	Format:	U14	Value	Description	0-16383	1-16K Pixels (encoder only goes to 8K)
	Format:	U14					
	Value	Description					
0-16383	1-16K Pixels (encoder only goes to 8K)						
<p>15:14 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ						
	<p>13:0 Altref Frame Width In Pixels Minus 1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U14</td> </tr> </table> <p>Specifies the width of the Last picture in units of pixels. The Altref picture width in units of luma samples equals (LastFrameWidthtInMinBlocksMinus1 + 1)</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-16383</td> <td>1-16K Pixels (encoder only goes to 8K)</td> </tr> </tbody> </table>	Format:	U14	Value	Description	0-16383	1-16K Pixels (encoder only goes to 8K)
	Format:	U14					
	Value	Description					
0-16383	1-16K Pixels (encoder only goes to 8K)						
<p>15:8 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ						
10	<p>31:16 First Partition Size in Bytes [15:0]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U16</td> </tr> </table> <p>Specifies the number of bytes taken up by the first partition size which handle the probability updates</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>Only used by Decoder</p>	Format:	U16	Programming Notes			
	Format:	U16					
	Programming Notes						
<p>7:0 Uncompressed Header Length in Bytes [7:0]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U8</td> </tr> </table> <p>Specifies the number of bytes taken up by the uncompressed frame header.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>this field is used by decoder only</p>	Format:	U8	Programming Notes				
Format:	U8						
Programming Notes							
<p>15:8 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ						
11	<p>31:4 ChickenBits [31:4]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table> <p>This field is reserved to be used for chicken bits</p>	Format:	MBZ				
	Format:	MBZ					
<p>3 Chroma Fractional Calculation Modified</p> <p>This changes how Chroma Fractional Calculation is handled in MC "0"--Default version (current); "1"--Matched an earlier version of ref decoder</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Value	Name					
Value	Name						



HCP_VP9_PIC_STATE										
	<table border="1"> <tr> <td>0</td> <td>Current [Default]</td> </tr> <tr> <td>1</td> <td>Outdated</td> </tr> </table>	0	Current [Default]	1	Outdated					
0	Current [Default]									
1	Outdated									
2	<p>MV Clamp Disable This modifies how hardware handles MV clamping logic "0"--MV clamping logic is enabled; "1"--MV clamping logic is disabled Not used in Encoder Mode</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Enable [Default]</td> </tr> <tr> <td>1</td> <td>Disable</td> </tr> </tbody> </table>	Value	Name	0	Enable [Default]	1	Disable			
Value	Name									
0	Enable [Default]									
1	Disable									
1	<p>Motion Comp Scaling Chicken Bit This modifies how VP9 Motion Comp to handle scaling x/y calculation. 0 -- Pre-Dec13 version; 1--Post-Dec13 version</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Pre-Dec13 Version</td> </tr> <tr> <td>1</td> <td>Post-Dec13 version [Default]</td> </tr> </tbody> </table>	Value	Name	0	Pre-Dec13 Version	1	Post-Dec13 version [Default]			
Value	Name									
0	Pre-Dec13 Version									
1	Post-Dec13 version [Default]									
0	<p>ChickenBits[0]</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>This field is reserved to be used for chicken bits</p>	Format:	MBZ							
Format:	MBZ									
12	<p>31:0 ChickenBits [63:32]</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Only valid value</td> </tr> <tr> <td>0x1-0xFFFFFFFF</td> <td>Invalid</td> </tr> </tbody> </table>	Format:	MBZ	Value	Description	0	Only valid value	0x1-0xFFFFFFFF	Invalid	
Format:	MBZ									
Value	Description									
0	Only valid value									
0x1-0xFFFFFFFF	Invalid									
13	<p>31:26 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>23:16 Base Q Index (Same as Luma AC)</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Added to Delta Q index of Segment Valid Values : 0..255</p> <table border="1"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> <tr> <td>Encoder Only</td> </tr> </table> <p>15:0 Compressed header BIN count</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>Number of bins compressed header This field is fixed insideHW</p> <table border="1"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table>	Format:	MBZ	Format:	U8	Programming Notes	Encoder Only	Format:	U16	Programming Notes
Format:	MBZ									
Format:	U8									
Programming Notes										
Encoder Only										
Format:	U16									
Programming Notes										



HCP_VP9_PIC_STATE			
		Encoder Only	
14	31:21	Reserved Format: MBZ	
	20:16	Luma DC Q Index Delta Format: S4 QP delta value for Luma DC Valid Values : -15..15 Programming Notes Encoder Only	
	15:13	Reserved Format: MBZ	
	12:8	ChromaDC_QindexDelta Format: S4 QP Delta Value For Chroma DC Valid Values : -15..15 Programming Notes Encoder Only	
	7:5	Reserved Format: MBZ	
	4:0	ChromaAC_QindexDelta Format: S4 QP Delta Value For Chroma AC Valid Values : -15..15 Programming Notes Encoder Only	
	15	31	Reserved Format: MBZ
		30:24	LF_ref_delta3 Format: S6 Loop filter level delta3 value; valid range -63...63 With this range -63..63, in the case of Base LF Level > 31 (first pass) or (Base LF Level + BRC accumulated LF Delta) > 31 (subsequent pass), the lf_ref_delta0,1,2,3 and lf_mode_delta0,1 in compressed bitstream would be in the range -31..31. Programming Notes Encoder Only
23		Reserved Format: MBZ	



HCP_VP9_PIC_STATE	
22:16	LF_ref_delta2 Format: S6 Loop filter level delta2 value; valid range -63..63 With this range -63..63, in the case of Base LF Level > 31 (first pass) or (Base LF Level + BRC accumulated LF Delta) > 31 (subsequent pass), the lf_ref_delta0,1,2,3 and lf_mode_delta0,1 in compressed bitstream would be in the range -31..31. Programming Notes Encoder Only
	15 Reserved Format: MBZ
	14:8 LF_ref_delta1 Format: S6 Loop filter level delta1 value; valid range -63..63 With this range -63..63, in the case of Base LF Level > 31 (first pass) or (Base LF Level + BRC accumulated LF Delta) > 31 (subsequent pass), the lf_ref_delta0,1,2,3 and lf_mode_delta0,1 in compressed bitstream would be in the range -31..31. Programming Notes Encoder Only
	7 Reserved Format: MBZ
6:0	LF_ref_delta0 Format: S6 Loop filter level delta0 value; valid range -63..63 With this range -63..63, in the case of Base LF Level > 31 (first pass) or (Base LF Level + BRC accumulated LF Delta) > 31 (subsequent pass), the lf_ref_delta0,1,2,3 and lf_mode_delta0,1 in compressed bitstream would be in the range -31..31. Programming Notes Encoder Only
	16 Reserved Format: MBZ
14:8	LF Mode Delta 1 Format: S6 Loop filter level mode1 value; valid range -63..63 With this range -63..63, in the case of Base LF Level > 31 (first pass) or (Base LF Level + BRC accumulated LF Delta) > 31 (subsequent pass), the lf_ref_delta0,1,2,3 and lf_mode_delta0,1 in compressed bitstream would be in the range -31..31. Programming Notes Encoder Only
	31:15 Reserved Format: MBZ



HCP_VP9_PIC_STATE				
	7	Reserved		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ
Format:	MBZ			
	6:0	LF Mode Delta 0		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>S6</td> </tr> </table> <p>Loop filter level mode1 value; valid range -63..63 With this range -63..63, in the case of Base LF Level > 31 (first pass) or (Base LF Level + BRC accumulated LF Delta) > 31 (subsequent pass), the lf_ref_delta0,1,2,3 and lf_mode_delta0,1 in compressed bitstream would be in the range -31..31.</p>	Format:	S6
		Format:	S6	
		Programming Notes		
Encoder Only				
17	31:16	BitOffsetForLFModeDelta		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U16</td> </tr> </table> <p>Offset from starting position of output bitstream in bits where LFModeDelta should be inserted. In BRC mode, always insert LFModeDelta. (This implies uncompressed header should have: mode_ref_delta_enabled=1 and mode_ref_delta_update=1) and The uncompressed header starting from this offset BitOffsetForLFModeDelta has to have the following 16 bits format: (Start here: 1'b1, mode_delta_0[6:0], 1'b1, mode_delta_1[6:0]) and BitOffsetForLFModeDelta = BitOffsetForLRefDelta + 32. Encoder Only</p>	Format:	U16
	Format:	U16		
	15:0	BitOffsetForLRefDelta		
<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U16</td> </tr> </table> <p>Offset from starting position of output bitstream in bits where LRefDelta should be inserted. In BRC mode, always insert LRefDelta. (This implies uncompressed header should have: mode_ref_delta_enabled=1 and mode_ref_delta_update=1) and The uncompressed header starting from this offset BitOffsetForLRefDelta has to have the following 32 bits format: (Start here: 1'b1, ref_delta_0[6:0], 1'b1, ref_delta_1[6:0], 1'b1, ref_delta_2[6:0], 1'b1, ref_delta_3[6:0]). Encoder Only</p>		Format:	U16	
Format:	U16			
18	31:16	BitOffsetForLFLevel		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U16</td> </tr> </table> <p>Offset from starting position of output bitstream in bits where LFLevel should be inserted. Encoder Only</p>	Format:	U16
Format:	U16			
	15:0	BitOffsetForQindex		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U16</td> </tr> </table> <p>Offset from starting position of output bitstream in bits where Qindex should be inserted. Encoder Only</p>	Format:	U16
Format:	U16			



HCP_VP9_SEGMENT_STATE

HCP_VP9_SEGMENT_STATE			
Source:		VideoCS	
Length Bias:		2	
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
Default Value:		7h Codec/Engine Name	
Format:		OpCode	
Codec/Engine Name = HUC = Bh			
22:16	Media Instruction Command		
	Default Value:	32h HCP_VP9_SEGMENT_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	
	6h		
1	31:3	Reserved	
		Format:	MBZ
	2:0	Segment ID	
Format:	U3		
		The segment ID of the DWORDS following this one	
2	31:4	Reserved	
		Format:	MBZ
	3	Segment Reference Enabled	
Format:	Enable		
		For encoder: When "Segment Reference Enabled" is set to 1, Software (Kernel) needs to program all PUs of this segment accordingly. This means: CU record PU reframe0="Segment Reference" bit[2:1], reframe1=0, and interpret_comp=single.	



HCP_VP9_SEGMENT_STATE							
	<p>2:1 Segment Reference</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Indicates reference frame for this segment.</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">If the current frame is a KEY frame or INTRA_ONLY frame, this field should be set to INTRA for all segments.</td> </tr> </table>	Format:	U2	Programming Notes		If the current frame is a KEY frame or INTRA_ONLY frame, this field should be set to INTRA for all segments.	
	Format:	U2					
Programming Notes							
If the current frame is a KEY frame or INTRA_ONLY frame, this field should be set to INTRA for all segments.							
0	<p>Segment Skipped</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Indicates skip mode for this segment.</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">If set to 1, all delta coefficients and MVs within CU of this segment should be forced to zero in HW. No block less than 8x8 size allowed segmentSkipped If set to 0, skipcoeff_flag should be coded as normal</td> </tr> </table>	Format:	Enable	Programming Notes		If set to 1, all delta coefficients and MVs within CU of this segment should be forced to zero in HW. No block less than 8x8 size allowed segmentSkipped If set to 0, skipcoeff_flag should be coded as normal	
	Format:	Enable					
Programming Notes							
If set to 1, all delta coefficients and MVs within CU of this segment should be forced to zero in HW. No block less than 8x8 size allowed segmentSkipped If set to 0, skipcoeff_flag should be coded as normal							
3	<p>31:30 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ					
	<p>29:24 FilterLevelRef1Mode1</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder mode only</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Indicates final filter level for Ref1 (Last Frame) and Mode 1.</p>	Exists If:	//Decoder mode only	Format:	U6		
	Exists If:	//Decoder mode only					
	Format:	U6					
	<p>23:22 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ					
	<p>21:16 FilterLevelRef1Mode0</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder mode only</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Indicates final filter level for Ref1 (Last Frame) and Mode 0.</p>	Exists If:	//Decoder mode only	Format:	U6		
	Exists If:	//Decoder mode only					
	Format:	U6					
<p>15:14 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ						
<p>13:8 FilterLevelRef0Mode1</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder mode only</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Indicates final filter level for Ref0 (Last Frame) and Mode 1.</p>	Exists If:	//Decoder mode only	Format:	U6			
Exists If:	//Decoder mode only						
Format:	U6						
<p>7:6 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ						
<p>5:0 FilterLevelRef0Mode0</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder mode only</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Indicates final filter level for Ref0 (Last Frame) and Mode 0.</p>	Exists If:	//Decoder mode only	Format:	U6			
Exists If:	//Decoder mode only						
Format:	U6						



HCP_VP9_SEGMENT_STATE						
4	31:30	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	29:24	<p>FilterLevelRef3Mode1</p> <table border="1" style="width: 100%;"> <tr> <td>Exists If:</td> <td>//Decoder mode only</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Indicates final filter level for Ref3 (Last Frame) and Mode 1.</p>	Exists If:	//Decoder mode only	Format:	U6
	Exists If:	//Decoder mode only				
	Format:	U6				
	23:22	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	21:16	<p>FilterLevelRef3Mode0</p> <table border="1" style="width: 100%;"> <tr> <td>Exists If:</td> <td>//Decoder mode only</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Indicates final filter level for Ref3 (Last Frame) and Mode 0.</p>	Exists If:	//Decoder mode only	Format:	U6
	Exists If:	//Decoder mode only				
	Format:	U6				
15:14	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
13:8	<p>FilterLevelRef2Mode1</p> <table border="1" style="width: 100%;"> <tr> <td>Exists If:</td> <td>//Decoder mode only</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Indicates final filter level for Ref2 (Last Frame) and Mode 1.</p>	Exists If:	//Decoder mode only	Format:	U6	
Exists If:	//Decoder mode only					
Format:	U6					
7:6	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
5:0	<p>FilterLevelRef2Mode0</p> <table border="1" style="width: 100%;"> <tr> <td>Exists If:</td> <td>//Decoder mode only</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Indicates final filter level for Ref2 (Last Frame) and Mode 0.</p>	Exists If:	//Decoder mode only	Format:	U6	
Exists If:	//Decoder mode only					
Format:	U6					
5	31:16	<p>Luma AC Quant Scale (Decode mode Only)</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S15</td> </tr> </table> <p>Indicates final value of Luma AC Quantized Scale value.</p>	Format:	S15		
	Format:	S15				
15:0	<p>Luma DC Quant Scale (Decode mode Only)</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S15</td> </tr> </table> <p>Indicates final value of Luma DC Quantized Scale value.</p>	Format:	S15			
Format:	S15					



HCP_VP9_SEGMENT_STATE		
6	31:16	Chroma AC Quant Scale (Decode mode Only) Format: S15 Indicates final value of Chroma AC Quantized Scale value.
	15:0	Chroma DC Quant Scale (Decode mode Only) Format: S15 Indicates final value of Chroma DC Quantized Scale value.



HCP_WEIGHTOFFSET_STATE

HCP_WEIGHTOFFSET_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>The HCP is selected with the Media Instruction Opcode "7h" for all HCP Commands. Each HCP command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>This slice level command is issued in both the encoding and decoding processes, if the weighted_pred_flag or weighted_bipred_flag equals one. If zero, then this command is not issued. Weight Prediction Values are provided in this command. Only Explicit Weight Prediction is supported in encoder.</p> <p>For P-Slice, this command is issued only once together with HCP_REF_IDX_STATE Command for L0 list. For B-Slice, this command can be issued up to two times together with HCP_REF_IDX_STATE Command, one for L0 list and one for L1 list.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
		Default Value:	7h Codec/Engine Name
		Format:	OpCode
		Codec/Engine Name = HCP = 7h	
22:16	Media Instruction Command		
	Default Value:	13h HCP_WEIGHTOFFSET_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	
	20h	32	
1	31:1	Reserved	
		Format:	MBZ



HCP_WEIGHTOFFSET_STATE								
	0	RefPicListNum Format: U1						
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Reference Picture List 0</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Reference Picture List 1</td> </tr> </tbody> </table>	Value	Name	0	Reference Picture List 0	1	Reference Picture List 1
	Value	Name						
	0	Reference Picture List 0						
1	Reference Picture List 1							
2..17	31:16	Reserved Format: MBZ						
	15:8	luma_offset_IX[i] Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15. Valid only if explicit weighted prediction for luma is enabled, otherwise must be zero.						
		Programming Notes This (combined with its MSbyte above) shall be in the range of $-WpOffsetHalfRange_V - 1$, where $WpOffsetHalfRange_V = 1 \ll (high_precision_offsets_enabled_flag ? (BitDepth_V - 1) : 7)$						
	7:0	delta_luma_weight_IX[i] Format: S7						
		Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15. Valid only if explicit weighted prediction for luma is enabled, otherwise must be zero.						
		Programming Notes When <code>luma_weight_l0_flag[i]</code> is equal to 1, the value of <code>delta_luma_weight_l0[i]</code> shall be in the range of -128 to 127, inclusive.						
18..33	31:24	ChromaOffsetLX [i][1] Where X is the RefPicListNum and i is the list entry number 0 through 15. DW 18 corresponds to i=0, DW 33 corresponds to i=15. Valid only if explicit weighted prediction for chroma is enabled, otherwise must be zero.						
		Programming Notes This (combined with its MSbyte below) shall be in the range of $-WpOffsetHalfRange_C$ to $(WpOffsetHalfRange_C - 1)$, inclusive $WpOffsetHalfRange_C = 1 \ll (high_precision_offsets_enabled_flag ? (BitDepth_C - 1) : 7)$						



HCP_WEIGHTOFFSET_STATE		
23:16	delta_chroma_weight_IX[i][1]	
	Format: S7	
	Where X is the RefPicListNum and i is the list entry number 0 through 15. DW 18 corresponds to i=0, DW 33 corresponds to i=15.	
	Valid only if explicit weighted prediction for chroma is enabled, otherwise must be zero.	
	Programming Notes	
	This shall be in the range of -128 to 127, inclusive	
	15:8	ChromaOffsetLX[i][0]
		Where X is the RefPicListNum and i is the list entry number 0 through 15. DW 18 corresponds to i=0, DW 33 corresponds to i=15.
		Valid only if explicit weighted prediction for chroma is enabled, otherwise must be zero.
		Programming Notes
This (combined with its MSbyte below) shall be in the range of -WpOffsetHalfRangeC to (WpOffsetHalfRangeC - 1), inclusive WpOffsetHalfRangeC = 1 « (high_precision_offsets_enabled_flag ? (BitDepthC - 1) : 7)		
7:0	delta_chroma_weight_IX[i][0]	
	Format: S7	
	Where X is the RefPicListNum and i is the list entry number 0 through 15. DW 18 corresponds to i=0, DW 33 corresponds to i=15.	
	Valid only if explicit weighted prediction for chroma is enabled, otherwise must be zero.	
	Programming Notes	
	This shall be in the range of -128 to 127, inclusive	



HEVC_VP9_RDOQ_STATE

HEVC_VP9_RDOQ_STATE				
Source:		VideoCS		
Length Bias:		2		
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	3h	
		Format:	Opcode	
			PARALLEL_VIDEO_PIPE	
	28:27	Pipeline		
		Default Value:	2h	
		Format:	Opcode	
			MFX_COMMON	
26:23	Opcode			
	Default Value:	7h		
	Format:	Opcode		
		Codec/Engine Name = HCP = 7h		
22:21	SubOpA	Default Value:	0h	
		Format:	Opcode	
20:16	SubOpB	Default Value:	8h	
		Format:	Opcode	
15:12	Reserved	Format:	MBZ	
11:0	DWord Length	Total Length - 2		
		Value	Name	Description
		3Ch	DWORD_COUNT_n [Default]	Excludes DWord (0,1)
1	31	Chicken bit to disable HTQ performance fix0 set to disable performance optimizations done by doubling LNZ and OSR storage Set to 1, to go back to single LNZ and OSR (no optimization) Set to 0, to use double buffer LNZ and OSR		



HEVC_VP9_RDOQ_STATE			
	30	<p>Chicken bit to disable HTQ performance fix1 set to disable performance optimization done to save 1clk while switching from EBB to GTRAM storage. This is critical for IntraLoop performance Set to 1, disable optimization Set to 0, enable optimization</p>	
	29:0	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:
Format:	MBZ		
2	31:16	<p>Lambda_Intra_Luma1 Lambda value for Intra Luma component of QP=1 for HEVC Lambda value for Intra Luma in segment 1 for VP9</p>	
	15:0	<p>Lambda_Intra_Luma0 Lambda value for Intra Luma component of QP=0 for HEVC Lambda value for Intra Luma in segment 0 for VP9</p>	
3	31:16	<p>Lambda_Intra_Luma3 Lambda value for Intra Luma component of QP=3 for HEVC Lambda value for Intra Luma in segment 3 for VP9</p>	
	15:0	<p>Lambda_Intra_Luma2 Lambda value for Intra Luma component of QP=2 for HEVC Lambda value for Intra Luma in segment 2 for VP9</p>	
4	31:16	<p>Lambda_Intra_Luma5 Lambda value for Intra Luma component of QP=5 for HEVC Lambda value for Intra Luma in segment 5 for VP9</p>	
	15:0	<p>Lambda_Intra_Luma4 Lambda value for Intra Luma component of QP=4 for HEVC Lambda value for Intra Luma in segment 4 for VP9</p>	
5	31:16	<p>Lambda_Intra_Luma7 Lambda value for Intra Luma component of QP=7 for HEVC Lambda value for Intra Luma in segment 7 for VP9</p>	
	15:0	<p>Lambda_Intra_Luma6 Lambda value for Intra Luma component of QP=6 for HEVC Lambda value for Intra Luma in segment 6 for VP9</p>	
6..33	31:16	<p>Lambda_Intra_Luma9,11, ...,61, 63(odd number) Lambda value for Intra Luma component of QP=9, 11, ..., 61, 63 (odd number) for HEVC</p>	
	15:0	<p>Lambda_Intra_Luma8, 10, ..., 48, 50 (even number) Lambda value for Intra Luma component of QP=8, 10, ..., 60, 62 (even number) for HEVC</p>	
34	31:16	<p>Lambda_Intra_Chroma 1 Lambda value for Intra Chroma component of QP=1 for HEVC Lambda value for Intra Chroma in segment 1 for VP9</p>	



HEVC_VP9_RDOQ_STATE		
	15:0	Lambda_Intra_Chroma 0 Lambda value for Intra Chroma component of QP=0 for HEVC'Lambda value for Intra Chroma in segment 0 for VP9
35	31:16	Lambda_Intra_Chroma 3 Lambda value for Intra Chroma component of QP=3 for HEVC'Lambda value for Intra Chroma in segment 3 for VP9
	15:0	Lambda_Intra_Chroma 2 Lambda value for Intra Chroma component of QP=2 for HEVC'Lambda value for Intra Chroma in segment 2 for VP9
36	31:16	Lambda_Intra_Chroma 5 Lambda value for Intra Chroma component of QP=5 for HEVC'Lambda value for Intra Chroma in segment 5 for VP9
	15:0	Lambda_Intra_Chroma 4 Lambda value for Intra Chroma component of QP=4 for HEVC'Lambda value for Intra Chroma in segment 4 for VP9
37	31:16	Lambda_Intra_Chroma 7 Lambda value for Intra Chroma component of QP=7 for HEVC'Lambda value for Intra Chroma in segment 7 for VP9
	15:0	Lambda_Intra_Chroma 6 Lambda value for Intra Chroma component of QP=6 for HEVC'Lambda value for Intra Chroma in segment 6 for VP9
38..65	31:16	Lambda_Intra_Chroma 9, 11, ..., 49, 51 (odd number) Lambda value for Intra Chroma component of QP=9, 11, ..., 61, 63 (odd number) for HEVC
	15:0	Lambda_Intra_Chroma 8, 10, ..., 48, 50 (even number) Lambda value for Intra Chroma component of QP=8, 10, ..., 60, 62 (even number) for HEVC
66	31:16	Lambda_Inter_Luma1 Lambda value for Inter Luma component of QP=1 for HEVC'Lambda value for Inter Luma in segment 1 for VP9
	15:0	Lambda_Inter_Luma0 Lambda value for Inter Luma component of QP=0 for HEVC'Lambda value for Inter Luma in segment 0 for VP9
67	31:16	Lambda_Inter_Luma3 Lambda value for Inter Luma component of QP=3 for HEVC'Lambda value for Inter Luma in segment 3 for VP9
	15:0	Lamba_Inter_Luma2 Lambda value for Inter Luma component of QP=2 for HEVC'Lambda value for Inter Luma in segment 2 for VP9
68	31:16	Lambda_Inter_Luma5 Lambda value for Inter Luma component of QP=5 for HEVC'Lambda value for Inter Luma in segment 5 for VP9



HEVC_VP9_RDOQ_STATE		
	15:0	Lambda_Inter_Luma4 Lambda value for Inter Luma component of QP=4 for HEVC'Lambda value for Inter Luma in segment 4 for VP9
69	31:16	Lambda_Inter_Luma7 Lambda value for Inter Luma component of QP=7 for HEVC'Lambda value for Inter Luma in segment 7 for VP9
	15:0	Lambda_Inter_Luma6 Lambda value for Inter Luma component of QP=6 for HEVC'Lambda value for Inter Luma in segment 6 for VP9
70..97	31:16	Lambda_Inter_Luma9, 11, ..., 49, 51 (odd number) Lambda value for Inter Luma component of QP= Luma9, 11, ..., 61, 63 (odd number) for HEVC
	15:0	Lambda_Intra_Inter 8, 10, ..., 48, 50 (even number) Lambda value for Inter Luma component of QP=8, 10, ..., 60, 62 (even number) for HEVC
98	31:16	Lambda_Inter_Chroma 1 Lambda value for Inter Chroma component of QP=1 for HEVC'Lambda value for Inter Chroma in segment 1 for VP9
	15:0	Lambda_Inter_Chroma0 Lambda value for Inter Chroma component of QP=0 for HEVC'Lambda value for Inter Chroma in segment 0 for VP9
99	31:16	Lambda_Inter_Chroma 3 Lambda value for Inter Chroma component of QP=3 for HEVC'Lambda value for Inter Chroma in segment 3 for VP9
	15:0	Lambda_Inter_Chroma 2 Lambda value for Inter Chroma component of QP=2 for HEVC'Lambda value for Inter Chroma in segment 2 for VP9
100	31:16	Lambda_Inter_Chroma 5 Lambda value for Inter Chroma component of QP=5 for HEVC'Lambda value for Inter Chroma in segment 5 for VP9
	15:0	Lambda_Inter_Chroma 4 Lambda value for Inter Chroma component of QP=4 for HEVC'Lambda value for Inter Chroma in segment 4 for VP9
101	31:16	Lambda_Inter_Chroma 7 Lambda value for Inter Chroma component of QP=7 for HEVC'Lambda value for Inter Chroma in segment 7 for VP9
	15:0	Lambda_Inter_Chroma 6 Lambda value for Inter Chroma component of QP=6 for HEVC'Lambda value for Inter Chroma in segment 6 for VP9
102..129	31:16	Lambda_Inter_Chroma 9, 11, ..., 49, 51 (odd number) Lambda value for Inter Chroma component of QP=9, 11, ..., 61, 63 (odd number) for HEVC
	15:0	Lambda_Inter_Chroma 8, 10, ..., 48, 50 (even number) Lambda value for Inter Chroma component of QP=8, 10, ..., 60, 62 (even number) for HEVC



HI8DS Render Target Write MSD

MSD_RTW_HI8DS - HI8DS Render Target Write MSD						
Source:	Render Cache DataPort					
Length Bias:	1					
Family:	Other					
Group:	Render Target R/W					
DWord	Bit	Description				
0	31	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
	Format:	MBZ				
	30	<p>Message Precision Subtype</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Full precision data message</p>	Default Value:	0h	Format:	Opcode
	Default Value:	0h				
	Format:	Opcode				
	29	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
	Format:	MBZ				
	28:25	<p>Message Length</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.</p>	Format:	U4		
Format:	U4					
24:20	<p>Response Length</p> <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.</p>	Format:	U5			
Format:	U5					
19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the 2-register header.</p>	Format:	MDC_MHP			
Format:	MDC_MHP					
18	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					



MSD_RTW_HI8DS - HI8DS Render Target Write MSD					
17:14	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0Ch</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Render Target Write message</p>	Default Value:	0Ch	Format:	Opcode
Default Value:	0Ch				
Format:	Opcode				
13	<p>Per-Sample PS Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.</p> <p style="text-align: center;">Programming Notes</p> <p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE.</p> <p>When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_CO as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0.</p> <p>When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples).</p> <p>When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>	Format:	Enable		
Format:	Enable				
12	<p>Last Render Target Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.</p> <p style="text-align: center;">Programming Notes</p> <p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set. In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>	Format:	Enable		
Format:	Enable				
11	<p>Slot Group Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MDC_RT_SGS</td> </tr> </table> <p>This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.</p>	Format:	MDC_RT_SGS		
Format:	MDC_RT_SGS				



MSD_RTW_HI8DS - HI8DS Render Target Write MSD	
10:8	Render Target Message Subtype
	Default Value: 3h
	Format: Opcode
	SIMD8 dual source message. Use slots [15:8] for pixel enables, X/Y addresses, and oMask.
	Programming Notes The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [31:24] are referenced instead of [15:8].
7:0	Binding Table Index
	Format: MDC_BT Specifies the Binding Table Index for the message



HUC_CFG_STATE

HUC_CFG_STATE		
Source:	VideoCS	
Length Bias:	2	
<p>The HUC is selected with the Media Instruction Opcode "Bh" for all HUC Commands. Each HUC command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HUC_CFG_STATE command is used to force the P24C (Minutela) into a reset condition as well as forcing it out of a reset condition. This command is not normally required since the hardware will handle placing the P24C into a reset condition and releasing it from reset, but there may be conditions that require a forced reset.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
		Format: OpCode
	28:27	Pipeline Type
		Default Value: 2h
		Format: OpCode
	26:23	Media Instruction Opcode
Default Value: Bh Codec/Engine Name		
Format: OpCode		
Codec/Engine Name = HUC = Bh		
22:16	Media Instruction Command	
	Default Value: 3h HUC_CFG_STATE	
	Format: OpCode	
15:12	Reserved	
	Format: MBZ	
11:0	Dword Length	
	Format: =n	
	(Excludes Dwords 0, 1).	
	Value	Name
	0h	
1	31:1	Reserved
		Format: MBZ



HUC_CFG_STATE											
	0	P24C (MinutelA)									
		Format: U1									
		<table border="1"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>0</td><td>Normal operation</td><td>No reset.</td></tr><tr><td>1</td><td>Force reset</td><td></td></tr></tbody></table>	Value	Name	Description	0	Normal operation	No reset.	1	Force reset	
		Value	Name	Description							
0	Normal operation	No reset.									
1	Force reset										



HUC_DMEM_STATE

HUC_DMEM_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>The HUC is selected with the Media Instruction Opcode "Bh" for all HUC Commands. Each HUC command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HUC_DMEM_STATE command is used to fetch the HUC data from the graphics memory and load it into the HUC 96KB L2 storage RAM. The HUC_DMEM_STATE specifies the data source base address in graphics memory.</p> <p>When the HUC_DMEM_STATE command is received, the data is loaded by the HUC DMA into the 96KB L2 storage RAM at the location provided in the HUC_DMEM_STATE command. This command also specifies the length of the data, which is specified in bytes but must be in increments of 64 byte cache lines.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
		Default Value:	Bh Codec/Engine Name
		Format:	OpCode
		Codec/Engine Name = HUC = Bh	
22:16	Media Instruction Command		
	Default Value:	2h HUC_DMEM_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	
	4h		
1..2	63:0	HUC Data Source Base Address	
		Format:	SplitBaseAddress64ByteAligned
Specifies the 64 byte aligned HUC data source base address in graphics memory.			



HUC_DMEM_STATE		
3	31:0	HUC Data Source
4	31:17	Reserved Format: _____ MBZ
	16:6	HUC Data Destination Base Address Format: _____ GraphicsAddress[16:6] Specifies the HUC Data destination base address in the L2 storage RAM. The base address is 64 byte cache aligned.
	5:0	Reserved Format: _____ MBZ
5	31:17	Reserved Format: _____ MBZ
	16:6	HUC Data Length Format: _____ GraphicsAddress[16:6] Specifies the length in bytes of the HUC Data. The length must be in increments of 64 byte cache lines.
	5:0	Reserved Format: _____ MBZ



HUC_IMEM_STATE

HUC_IMEM_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>The HUC is selected with the Media Instruction Opcode "Bh" for all HUC Commands. Each HUC command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HUC_IMEM_STATE command is used to fetch the HUC firmware from the WOPCM region and load it into the HUC 96KB L2 storage RAM. The HUC_IMEM_STATE specifies the firmware's offset in WOPCM which is a cache line aligned 32-bit offset address. The firmware, code and data segment size should always be less than 96KB. If its greater than 96KB then the firmware should be broken into smaller functions so that there is no firmware straddling case.</p> <p>When the HUC_IMEM_STATE command is received, the hardware confirms that the code has been successfully authenticated by checking the VCR provided authentication successful signal. If this signal is asserted, the firmware is loaded by the HUC DMA into the 96KB L2 storage RAM. Once the firmware is loaded, the VALID IMEM LOADED bit in the HUC_STATUS2 register is asserted high in the. If the authentication signal is not asserted, the DMA aborts the HUC_IMEM_STATE command, the firmware is not loaded, and the VALID IMEM LOADED bit remains low.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
		Default Value:	Bh Codec/Engine Name
		Format:	OpCode
			Codec/Engine Name = HUC = Bh
22:16	Media Instruction Command		
	Default Value:	1h HUC_IMEM_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	
	3h		



HUC_IMEM_STATE										
1	31:0	Reserved								
		Format: MBZ								
2	31:0	Reserved								
		Format: MBZ								
3	31:0	Reserved								
		Format: MBZ								
4	31:8	Reserved								
	7:0	HUC Firmware Descriptor								
		Format: U8								
		This field specifies 1 of 255 firmware descriptors which describe which firmware is be loaded in the L2 storage RAM. If the firmware descriptor is set to zero, the HUC will not load the firmware.								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>Illegal</td> </tr> <tr> <td style="text-align: center;">[1,255]</td> <td></td> <td>Firmware Descriptor</td> </tr> </tbody> </table>		Value	Name	Description	0		Illegal	[1,255]		Firmware Descriptor
Value	Name	Description								
0		Illegal								
[1,255]		Firmware Descriptor								



HUC_IND_OBJ_BASE_ADDR_STATE

HUC_IND_OBJ_BASE_ADDR_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>The HUC is selected with the Media Instruction Opcode "Bh" for all HUC Commands. Each HUC command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HUC_IND_OBJ_BASE_ADDR_STATE command is used to define the indirect object base address of the stream in graphics memory. This is a frame level command.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
	26:23	Media Instruction Opcode	
		Default Value:	Bh Codec/Engine Name
Format:		OpCode	
Codec/Engine Name = HUC = Bh			
22:16	Media Instruction Command		
	Default Value:	5h HUC_IND_OBJ_BASE_ADDR_STATE	
15:12	Reserved		
	Format:	MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
1..2	63:0	HUC Indirect Stream In ObjectBase Address	
		Format: SplitBaseAddress4KByteAligned	
Specifies the 4K-byte aligned memory base address for the read-only indirect data object pointed in the HUC_STREAM_OBJECT command for fetching (reading) the bit stream data.			
3	31:0	HUC Indirect Stream In ObjectBase Attributes	
		Format: MemoryAddressAttributes	



HUC_IND_OBJ_BASE_ADDR_STATE		
6..7	63:0	HUC Indirect Stream Out ObjectBase Address Format: SplitBaseAddress4KByteAligned Specifies the 4K-byte aligned memory base address for the indirect data object pointed in the HUC_STREAM_OBJECT command for writing the bit stream data.
8	31:0	HUC Indirect Stream Out ObjectBase Attributes Format: MemoryAddressAttributes
9..10	63:0	HUC Indirect Stream Out ObjectAccess Upper Bound Format: SplitBaseAddress4KByteAligned This field specifies the 4K-byte aligned maximum memory address access by the indirect data object in the HUC_STREAM_OBJECT command for the output bit stream. Indirect data accessed at this address or greater will cause the HUC to stop issuing requests to the GAC. Setting this field to 0 will cause this range to be ignored by the HUC.



HUC_PIPE_MODE_SELECT

HUC_PIPE_MODE_SELECT			
Source:	VideoCS		
Length Bias:	2		
<p>The HUC is selected with the Media Instruction Opcode "Bh" for all HUC Commands. Each HUC command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HUC_PIPE_MODE_SELECT command is responsible for general pipeline level configuration that would normally be set once for a single stream decode and would not be modified on a frame workload basis.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
	26:23	Media Instruction Opcode	
		Default Value:	Bh Codec/Engine Name
Format:		OpCode	
Codec/Engine Name = HUP = Bh			
22:16	Media Instruction Command		
	Default Value:	0h HUC_PIPE_MODE_SELECT	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	
1	31:24	Reserved	
	23:11	Reserved	
		Format:	MBZ
9:5	Reserved		
	Format:	MBZ	



HUC_PIPE_MODE_SELECT							
4	Indirect Stream Out Enable Format: _____ Enable Enables the bitstream to be written out to memory immediately following the decryption of the bit stream. The memory buffer is addressed through the HuC Indirect Stream Out ObjectBase Address. <table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>0h</td><td>Disable Indirect Stream Out</td></tr><tr><td>1h</td><td>Enable Indirect Stream Out</td></tr></tbody></table>	Value	Name	0h	Disable Indirect Stream Out	1h	Enable Indirect Stream Out
	Value	Name					
0h	Disable Indirect Stream Out						
1h	Enable Indirect Stream Out						
3:0	Reserved Format: _____ MBZ						
2	31:0 Media Soft-Reset Counter (per 1000 clocks) Format: _____ U32 In decoder modes, this counter value specifies the number of clocks (per 1000) of GAC inactivity before a media soft-reset is applied to the HCP and HuC. If counter value is set to 0, the media soft-reset feature is disabled and no reset will occur. In encoder modes, this counter must be set to 0 to disable media soft reset. This feature is not supported for the encoder. <table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>0</td><td>Disable</td></tr></tbody></table>	Value	Name	0	Disable		
	Value	Name					
0	Disable						



HUC_START

HUC_START			
Source:	VideoCS		
Length Bias:	2		
The HUC is selected with the Media Instruction Opcode "Bh" for all HUC Commands. Each HUC command has assigned a media instruction command as defined in DWord 0, BitField 22:16.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline Type	
		Default Value:	2h
		Format:	OpCode
	26:23	Media Instruction Opcode	
Default Value:		Bh Codec/Engine Name	
Format:		OpCode	
Codec/Engine Name = HUC = Bh			
22:16	Media Instruction Command		
	Default Value:	21h HUC_START	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	Dword Length		
	Format:	=n	
	(Excludes Dwords 0, 1).		
	Value	Name	
0h			
1	31:3	Reserved	
		Format:	MBZ
	2:1	Reserved	
	0	LastStreamObject	
Format:		U1	
Value		Name	Description
1		LastStreamObject	Last stream object in the workload.
0	NotLastStreamObject	Not the last stream object in the workload.	



HUC_STREAM_OBJECT

HUC_STREAM_OBJECT				
Source:	VideoCS			
Length Bias:	2			
<p>The HUC is selected with the Media Instruction Opcode "Bh" for all HUC Commands. Each HUC command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HUC_STREAM_OBJECT command is used to define the bit stream address offset to the Stream Indirect Object base Address and the length of the bit stream. The bitstream buffer the HUC operates upon is specified through indirect addressing.</p>				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	3h PARALLEL_VIDEO_PIPE	
		Format:	OpCode	
	28:27	28:27	Pipeline Type	
			Default Value:	2h
			Format:	OpCode
	26:23	26:23	Media Instruction Opcode	
			Default Value:	Bh Codec/Engine Name
			Format:	OpCode
			Codec/Engine Name = HUC = Bh	
	22:16	22:16	Media Instruction Command	
			Default Value:	20h HUC_STREAM_OBJECT
Format:			OpCode	
15:12	15:12	Reserved		
		Format:	MBZ	
11:0	11:0	Dword Length		
		Format:	=n	
		(Excludes Dwords 0, 1).		
		Value	Name	
		3h		



HUC_STREAM_OBJECT																	
1	31:0	Indirect Stream In Data Length															
		Format: U32															
		Specifies the length in bytes of the bit stream input data.															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[268435456,2147483547]</td> <td>Data_Length_beyond_28_bits</td> <td>This is an added valid range when the bit length is extended from 28 bits to 32 bits. This is added to support 16k x 16k picture size bitstream.</td> </tr> </tbody> </table>	Value	Name	Description	[268435456,2147483547]	Data_Length_beyond_28_bits	This is an added valid range when the bit length is extended from 28 bits to 32 bits. This is added to support 16k x 16k picture size bitstream.									
Value	Name	Description															
[268435456,2147483547]	Data_Length_beyond_28_bits	This is an added valid range when the bit length is extended from 28 bits to 32 bits. This is added to support 16k x 16k picture size bitstream.															
2	31	Reserved															
	30:29	Reserved															
		Format: MBZ															
28:0	Indirect Stream In Start Address																
	Format: U29																
Specifies the byte-aligned graphics memory starting address of the input bit stream relative to the HUC Indirect Stream In ObjectBase Address [31:12] .																	
3	31:29	Reserved															
		Format: MBZ															
	28:0	Indirect Stream Out Start Address															
Format: U29																	
Specifies the byte-aligned graphics memory starting address of the output bit stream relative to the HUC Indirect Stream Out ObjectBase Address [31:12] .																	
4	31:30	Reserved															
		Format: MBZ															
	29	HuC Bitstream Enable															
		Format: Enable															
		Enables the bitstream to be sent to the HuC															
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> </tr> <tr> <td>1h</td> <td>Enable</td> </tr> </tbody> </table>		Value	Name	0h	Disable	1h	Enable										
Value	Name																
0h	Disable																
1h	Enable																
28:27	DRMLengthMode																
	Format: U2																
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Start Code Mode</td> <td>Stops on a start code</td> </tr> <tr> <td>01b</td> <td>Length Mode</td> <td>Stops after a number of bytes are reached in the length counter</td> </tr> <tr> <td>10b</td> <td>Reserved</td> <td></td> </tr> <tr> <td>11b</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>		Value	Name	Description	00b	Start Code Mode	Stops on a start code	01b	Length Mode	Stops after a number of bytes are reached in the length counter	10b	Reserved		11b	Reserved	
	Value	Name	Description														
	00b	Start Code Mode	Stops on a start code														
01b	Length Mode	Stops after a number of bytes are reached in the length counter															
10b	Reserved																
11b	Reserved																



HUC_STREAM_OBJECT

	26	Stream Out	Format: Enable	
	Enables the serpent encrypted stream output.			
		Value	Name	Description
		0	Disable	Disable the serpent encrypted stream output.
		1	Enable	Enable the serpent encrypted stream output.
	25	Emulation Prevention Byte Removal	Format: Enable	
	Enables the serpent encrypted stream output.			
		Value	Name	Description
		0	Disable	Bypass Emulation Prevention Byte Removal.
		1	Enable	Emulation prevention bytes will be removed after the start code search engine.
24	Start Code Search Engine	Format: Enable		
Enables the serpent encrypted stream output.				
	Value	Name	Description	
	0	Disable	Bypass Start Code Search Engine	
	1	Enable	Enables the start code search engine to stop on every third byte start code defined by Start Code Byte [2:0] defined in this DWord.	
23:16	Start Code Byte [2]	Format: U8		
Third byte of the start code				
15:8	Start Code Byte [1]	Format: U8		
Second byte of the start code				
7:0	Start Code Byte [0]	Format: U8		
First byte of the start code				



HUC_VIRTUAL_ADDR_STATE

HUC_VIRTUAL_ADDR_STATE						
Source:	VideoCS					
Length Bias:	2					
<p>HUC is selected with the Media Instruction Opcode "Bh" for all HUC Commands. Each HUC command has assigned a media instruction command as defined in DWord 0, BitField 22:16.</p> <p>The HUC_VIRTUAL_ADDR_STATE command is used to define the 48-bit HUC Surface Base Address and HUC Surface for each region.</p>						
DWord	Bit	Description				
0	31:29	Command Type Default Value: 3h PARALLEL_VIDEO_PIPE Format: OpCode				
	28:27	Pipeline Type Default Value: 2h Format: OpCode				
	26:23	Media Instruction Opcode Default Value: Bh Codec/Engine Name Format: OpCode Codec/Engine Name = HUC = Bh				
	22:16	Media Instruction Command Default Value: 4h HUC_VIRTUAL_ADDR_STATE Format: OpCode				
	15:12	Reserved Format: MBZ				
	11:0	Dword Length Format: =n (Excludes Dwords 0, 1). <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">2Fh</td> <td></td> </tr> </tbody> </table>	Value	Name	2Fh	
	Value	Name				
2Fh						
1..48 Regions 0 through 15 use three DWords each to describe the HUC Surface Base Address and HUC Surface, for a total of 16 regions x 3 DWords/each = 48 DWords.	95:64	HUC Surface (VirtualAddrRegion[0-15])				
	63:0	HUC Surface Base Address (VirtualAddrRegion[0-15]) Format: SplitBaseAddress4KByteAligned HUC surface base address for each virtual address region [0-15].				



If

if - If			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>An if instruction starts an if/endif or an if/else/endif block of code. It restricts execution within the conditional block to only those channels that were enabled via the predicate control.</p> <p>Each if instruction must have a matching endif instruction and may have up to one matching else instruction before the matching endif.</p> <p>If all channels are inactive (for the if/endif or if/else/endif block), a jump is performed to the instruction referenced by JIP. This jump must be to right after the matching else instruction when present, or otherwise to the matching endif instruction of the conditional block.</p> <p>If SPF is ON, the UIP must be used to update IP; JIP is not used in this case.</p> <p>The following table describes the 32-bit exit code <JIP> and <UIP>. If <branch_ctrl> is set, then the JIP points to the first join instruction within the if block. If <branch_ctrl> is not set, <JIP> should point to the instruction right after the matching else instruction if it exists, otherwise <JIP> should point to the endif instruction. <UIP> should always point to the endif instruction. When a jump occurs, this value is added to IP pre-increment. In GEN instruction binary, <JIP> and <UIP> are at location <src0> & <src1> and must be of type D (signed dword integer).</p>			
Format: [(pred)] if (exec_size JIP UIP <branch_ctrl>)			
Restriction			
Restriction : The execution size must be the same for the if, else, and endif instructions of the same code block.			
Syntax			
[(pred)] if (exec_size) imm32 imm32 <branch_ctrl>			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < 32; n++) { if (WrEn.channel[n] == 0) { PcIP[n] = IP + JIP; } else { PcIP[n] = IP + 1; } } if (PcIP != (IP + 1)) { // for all channels Jump(IP + JIP); }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
DWord	Bit	Description	
0..3	127:96	JIP Format: S31 The byte-aligned jump distance if a jump is taken for the channel.	



if - If				
	95:64	<p>UIP</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S31</td> </tr> </table> <p>The byte aligned jump distance if a jump is taken for the instruction.</p>	Format:	S31
	Format:	S31		
	63:32	<p>Operand Control</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">EU_INSTRUCTION_OPERAND_CONTROLS</td> </tr> </table>	Format:	EU_INSTRUCTION_OPERAND_CONTROLS
Format:	EU_INSTRUCTION_OPERAND_CONTROLS			
31:0	<p>Header</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">EU_INSTRUCTION_HEADER</td> </tr> </table>	Format:	EU_INSTRUCTION_HEADER	
Format:	EU_INSTRUCTION_HEADER			



Illegal

illegal - Illegal			
Source:	Eulsa		
Length Bias:	4		
<p>The Illegal Opcode Exception Enable flag in cr0.1 is normally set so the normal processing of an illegal opcode is to transfer control to the System Routine.</p> <p>Instruction dispatch treats any unused 8-bit opcode (including bit 7 of the instruction, reserved for future opcode expansion) as if it is the illegal opcode.</p> <p>The illegal opcode is zero because that byte value is more likely than most to be read via a wayward instruction pointer.</p> <p>The illegal instruction is an instruction only in the same way that a NULL pointer in software is a pointer. Both are special values indicating invalid instances.</p>			
Format:	illegal		
Restriction			
Restriction : The illegal instruction takes no instruction options.			
Syntax			
illegal			
Pseudocode			
<pre>{ Set the Illegal Opcode Exception Status bit in cr0.1. if (Illegal Opcode Exception Enable is set in cr0.1) { Transfer control to the System Routine (return address to AIP, IP = SIP). } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
N	N	N	N
DWord	Bit	Description	
0..3	127:7	Reserved	
		Format:	MBZ
	6:0	Opcode	
		Format:	EU_OPCODE



Integer Subtraction with Borrow

subb - Integer Subtraction with Borrow			
Source:	Eulsa		
Length Bias:	4		
<p>The subb instruction performs component-wise subtraction of src0 and src1 and stores the results in dst, it also stores the borrow into acc. If the operation produces a borrow (src0 < src1), write 0x00000001 to acc, else write 0x00000000 to acc.</p>			
Format:	[(pred)] subb[.cmod] (exec_size) dst src0 src1		
Restriction			
Restriction : AccWrEn is required. The accumulator is an implicit destination and thus cannot be an explicit destination operand.			
Syntax			
[(pred)] subb[.cmod] (exec_size) reg reg reg [(pred)] subb[.cmod] (exec_size) reg reg imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src0.chan[n] - src1.chan[n]; acc.chan[n] = borrow(src.chan[n] - src1.chan[n]); } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	Y	N
Src Types	Dst Types		
UD	UD		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource	
Exists If:		([ImmSource][Src1.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_REG_IMM		
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Join

join - Join			
Source:	Eulsa		
Length Bias:	4		
<p>The join instruction makes the inactive channels active at the join IP if those channels are predicated. Any deactivated channels due to a goto instruction match the join IP are activated (qualified with predicates at join). If no IP is matched at this join, the program goes to the next IP with the active channels which followed the program path up to the join instruction. If no active channels are present after executing the join instruction, the program jumps to the offset specified by JIP instead of next IP.</p> <p>The join instruction is used in conjunction with a goto instruction. The join activates channels that are deactivated by the goto instruction. See the goto instruction for the deactivation rules.</p> <p>The goto and join instructions enable unstructured program control flow. These instructions must be used with additional care where dangling channels can result without proper compiler checks, meaning that it is expected that programs will navigate through these paths to reactivate the channels. Hardware does not provide native checks or reconvergence.</p> <p>The following table describes the 32-bit JIP. In GEN binary, JIP is at location src1 and must be of type D (signed DWord integer). JIP must be an immediate operand and is a signed 32-bit number. This value is added to IP pre-increment.</p> <p>If SPF is ON, none of the PcIP are updated.</p>			
Format:	[(pred)] join (exec_size) JIP		
Programming Notes			
An index of 0 does nothing, continuing execution with the next instruction.			
An index of -16 (if the jmpip instruction is in native format) or -8 (if the jmpip instruction is in compact format) is an infinite loop on the jmpip instruction.			
Restriction			
Restriction : The {NoMask} instruction option must be specified.			
Restriction : The index data type must be D (Signed DWord Integer).			
Syntax			
[(pred)] join (exec_size) imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { // for the predicated channels and the remaining channels PcIP[n] = IP + 1; } } if (PcIP != (IP + 1)) { // for all channels when no channels are activated and no other active channels Jump(IP + JIP); }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier



join - Join			
Y	N	N	N
DWord	Bit	Description	
0..3	127:96	JIP Format: S31 Jump Target Offset. The relative offset in bytes if a jump is taken for the instruction.	
	95	Source 0 Address Immediate [9] Sign Bit	
	94:91	Src1.SrcType Format: SrcType	
	90:89	Src1.RegFile Format: RegFile	
	88:64	Source 0 Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16') Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16	
	88:64	Source 0 Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1') Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1	
	63:32	Operand Control Format: EU_INSTRUCTION_OPERAND_CONTROLS	
	31:0	Header Format: EU_INSTRUCTION_HEADER	



Jump Indexed

jmp - Jump Indexed			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>The jmp instruction redirects program execution to an index offset relative to the post-incremented instruction pointer. The index is a signed integer value, with positive or zero integers for forward jumps, and negative integers for backward jumps.</p> <p>Note: Unlike other flow control instructions, the offset used by jmp is relative to the incremented instruction pointer rather than the IP value for the instruction itself.</p> <p>In GEN binary, index is at location src1. The ip register must be put (for example, by the assembler) at the dst and src0 locations.</p> <p>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.</p>			
Format:			
[[pred]] jmp (1) index {NoMask}			
Programming Notes			
An index of 0 does nothing, continuing execution with the next instruction.			
An index of -16 (if the jmp instruction is in native format) or -8 (if the jmp instruction is in compact format) is an infinite loop on the jmp instruction.			
Restriction			
Restriction : The execution size must be 1.			
Restriction : The {NoMask} instruction option must be specified.			
Restriction : The index data type must be D (Signed DWord Integer).			
Restriction : QtrCtrl must not be used for jmp instruction.			
Syntax			
[[pred]] jmp (1) reg32 {NoMask} [[pred]] jmp (1) imm32 {NoMask}			
Pseudocode			
Evaluate(WrEn); if (WrEn != 0) { Jump(IP + 1 + index); // IP + 1 is a pseudocode idiom for the IP of the following instruction. }			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types			



jmp - Jump Indexed					
D					
DWord	Bit	Description			
0..3	127:96	<p>JIP</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S31</td> </tr> </table> <p>Jump Target Offset. The relative offset in bytes if a jump is taken for the instruction.</p>	Format:	S31	
	Format:	S31			
	95	Source 0 Address Immediate [9] Sign Bit			
	94:91	<p>Src1.SrcType</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>SrcType</td> </tr> </table>	Format:	SrcType	
	Format:	SrcType			
	90:89	<p>Src1.RegFile</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>RegFile</td> </tr> </table>	Format:	RegFile	
	Format:	RegFile			
	88:64	<p>Source 0</p> <table border="1" style="width: 100%;"> <tr> <td>Exists If:</td> <td>(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16')</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16</td> </tr> </table>	Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16')	Format:
Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16')				
Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16				
88:64	<p>Source 0</p> <table border="1" style="width: 100%;"> <tr> <td>Exists If:</td> <td>(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1')</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1</td> </tr> </table>	Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1')	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1
Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1')				
Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1				
63:32	<p>Operand Control</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>EU_INSTRUCTION_OPERAND_CONTROLS</td> </tr> </table>	Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
Format:	EU_INSTRUCTION_OPERAND_CONTROLS				
31:0	<p>Header</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>EU_INSTRUCTION_HEADER</td> </tr> </table>	Format:	EU_INSTRUCTION_HEADER		
Format:	EU_INSTRUCTION_HEADER				



Leading Zero Detection

lzd - Leading Zero Detection			
Source:	Eulsa		
Length Bias:	4		
The lzd instruction counts component-wise the leading zeros from src0 and stores the resulting counts in dst. If src0 is zero, store 32 in dst.			
Format:	[(pred)] lzd[.cmod] (exec_size) dst src0		
Syntax			
[(pred)] lzd[.cmod] (exec_size) reg reg [(pred)] lzd[.cmod] (exec_size) reg imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { UD udScalar = src0.chan[n]; UD cnt = 0; while ((udScalar & (1 << 31)) == 0 && cnt != 32) { cnt ++; udScalar = udScalar << 1; } dst.chan[n] = cnt; } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
D, UD	UD		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([Operand Controls][Src0.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource	
Exists If:		([Operand Controls][Src0.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Line

line - Line			
Source:	Eulsa		
Length Bias:	4		
<p>The line instruction computes a component-wise line equation ($v = p * u + q$ where u, v are vectors and p, q are scalars) of <code>src0</code> and <code>src1</code> and stores the results in <code>dst</code>. <code>src1</code> is the input vector u. <code>src0</code> provides input scalars p and q, where p is the scalar value based on the region description of <code>src0</code> and q is the scalar value implied from <code>src0</code> region. Specifically, q is the fourth component of the 4-tuple (128-bit aligned) that p belongs to.</p>			
Format:	[[pred]] line[.cmod] (exec_size) dst src0 src1		
Restriction			
Restriction : This is a specialized instruction that only supports an execution size (ExecSize) of 8 or 16.			
Restriction : The <code>src0</code> region must be a replicated scalar (with <code>HorzStride == VertStride == 0</code>).			
Restriction : <code>src0</code> must specify <code>.0</code> or <code>.4</code> 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			
<pre>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; } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource	
Exists If:		([ImmSource][Src1.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_REG_IMM		



line - Line		
	63:32	Operand Controls Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header Format: EU_INSTRUCTION_HEADER



Linear Interpolation

Irp - Linear Interpolation			
Source:	Eulsa		
Length Bias:	4		
The Irp 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)] Irp[.cmod] (exec_size) dst src0 src1 src2		
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 Formats.			
Syntax			
[(pred)] Irp[.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] * src0.chan[n] + src2.chan[n] * (1.0 - src0.chan[n]); } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:126	Reserved Format: MBZ	
	125:106	Source 2 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC	
	105	Reserved Format: MBZ	
	104:85	Source 1 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC	



Irp - Linear Interpolation												
84	Reserved	Format: MBZ										
83:64	Source 0	Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC										
63:56	Destination Register Number	Format: DstRegNum										
55:53	Destination Subregister 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 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 Type This field contains the data type for the destination	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Single Precision Float</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>DWord</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Unsigned DWord</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Double Precision Float</td> </tr> </tbody> </table>	Value	Name	00b	Single Precision Float	01b	DWord	10b	Unsigned DWord	11b	Double Precision Float
Value	Name											
00b	Single Precision Float											
01b	DWord											
10b	Unsigned DWord											
11b	Double Precision Float											
48:42	Reserved	Format: MBZ										
43:42	Source Data Type This field contains the data type for all three sources	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Single Precision Float</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>DWord</td> </tr> </tbody> </table>	Value	Name	00b	Single Precision Float	01b	DWord				
Value	Name											
00b	Single Precision Float											
01b	DWord											



Irp - Linear Interpolation		
	10b	Unsigned DWord
	11b	Double Precision Float
41:40	Source 2 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
39:38	Source 1 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
41:36	Reserved	
	Exists If:	/// ([Property[Source Modifier] == 'false')
	Format:	MBZ
37:36	Source 0 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
35	Reserved	
	Format:	MBZ
34	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	Destination Register File	
	Value	Name
	0	GRF
	1	MRF
32	Reserved	
	Format:	MBZ
31:0	Header	
	Format:	EU_INSTRUCTION_HEADER



LO8DS Render Target Write MSD

MSD_RTW_LO8DS - LO8DS Render Target Write MSD		
Source:	Render Cache DataPort	
Length Bias:	1	
Family:	Other	
Group:	Render Target R/W	
DWord	Bit	Description
0	31	Reserved
		Format: MBZ Ignored
	30	Message Precision Subtype
		Default Value: 0h
		Format: Opcode Full precision data message
	29	Reserved
		Format: MBZ Ignored
	28:25	Message Length
Format: U4 Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.		
24:20	Response Length	
	Format: U5 Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.	
19	Header Present	
	Format: MDC_MHP If set, indicates that the message includes the 2-register header.	
18	Reserved	
	Format: MBZ Ignored	



MSD_RTW_LO8DS - LO8DS Render Target Write MSD					
17:14	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0Ch</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Render Target Write message</p>	Default Value:	0Ch	Format:	Opcode
Default Value:	0Ch				
Format:	Opcode				
13	<p>Per-Sample PS Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.</p> <p style="text-align: center;">Programming Notes</p> <p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE. When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_C0 as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0. When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples). When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case. This bit MUST be 0 for Dual Source Messages.</p>	Format:	Enable		
Format:	Enable				
12	<p>Last Render Target Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.</p> <p style="text-align: center;">Programming Notes</p> <p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set. In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>	Format:	Enable		
Format:	Enable				
11	<p>Slot Group Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MDC_RT_SGS</td> </tr> </table> <p>This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.</p>	Format:	MDC_RT_SGS		
Format:	MDC_RT_SGS				
10:8	<p>Render Target Message Subtype</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>2h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>SIMD8 dual source message. Use slots [7:0] for pixel enables, X/Y addresses, and oMask.</p>	Default Value:	2h	Format:	Opcode
Default Value:	2h				
Format:	Opcode				



MSD_RTW_LO8DS - LO8DS Render Target Write MSD							
	<table border="1"><thead><tr><th colspan="2">Programming Notes</th></tr></thead><tbody><tr><td colspan="2">The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [23:16] are referenced instead of [7:0].</td></tr></tbody></table>	Programming Notes		The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [23:16] are referenced instead of [7:0].			
Programming Notes							
The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [23:16] are referenced instead of [7:0].							
7:0	<table border="1"><thead><tr><th colspan="2">Binding Table Index</th></tr></thead><tbody><tr><td>Format:</td><td>MDC_BTS</td></tr><tr><td colspan="2">Specifies the Binding Table Index for the message</td></tr></tbody></table>	Binding Table Index		Format:	MDC_BTS	Specifies the Binding Table Index for the message	
Binding Table Index							
Format:	MDC_BTS						
Specifies the Binding Table Index for the message							



Logic And

and - Logic And			
Source:	Ealsa		
Length Bias:	4		
<p>The and instruction performs component-wise logic AND operation between src0 and src1 and stores the results in dst.</p> <p>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 AND operation. Any source modifier is logical, optionally changing a source value s to ~s (inverting all source bits). This capability allows expressions like a AND (NOT b) to be calculated with one instruction. This operation does not produce sign or overflow conditions. Only the .e/.z or .ne/.nz conditional modifiers should be used.</p>			
<p>Format: Source modifier is not allowed if source is an accumulator.</p>			
Restriction			
Restriction : Source modifier is not allowed if source is an accumulator.			
Syntax			
<pre>[(pred)] and[.cmod] (exec_size) reg reg reg [(pred)] and[.cmod] (exec_size) reg reg imm32</pre>			
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	Saturation	Source Modifier
Y	Y	N	Y
Src Types		Dst Types	
*B,*W,*D		*B,*W,*D	
*W,*D		*W,*D	
*W,*D		*Q	
*Q		*W,*D	
*Q		*Q	
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG



and - Logic And		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Logic Not

not - Logic Not			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>The not instruction performs logical NOT operation (or one's complement) of src0 and storing the results in dst.</p> <p>This operation does not produce sign or overflow conditions. Only the .e/z or .ne/.nz conditional modifiers should be used.</p> <p>A register source operand can use a source modifier:</p> <p>Any source modifier is logical, optionally changing a source value <i>s</i> to $\sim s$ (inverting all source bits). Such a source modifier is not particularly useful with the not instruction, as it changes the effect of not to just copying bits.</p>			
Format:	[[pred]] not[.cmod] (exec_size) dst src0		
Restriction			
Restriction : Source modifier is not allowed if source is an accumulator.			
Syntax			
[[pred]] not[.cmod] (exec_size) reg reg [[pred]] not[.cmod] (exec_size) reg imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = ~ src0.chan[n]; } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	N	Y
Src Types		Dst Types	
*B,*W,*D		*B,*W,*D	
*W,*D		*W,*D	
*W,*D		*Q	
*Q		*W,*D	
*Q		*Q	
DWord	Bit	Description	
0..3	127:64	RegSource Exists If: (([Operand Controls][Src0.RegFile]!='IMM') Format: EU_INSTRUCTION_SOURCES_REG	



not - Logic Not		
	127:64	ImmSource
		Exists If: ((Operand Controls)[Src0.RegFile]== 'IMM')
	Format: EU_INSTRUCTION_SOURCES_IMM32	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Logic Or

or - Logic Or			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>The or instruction performs component-wise logic OR operation between src0 and src1 and stores the results in dst.</p> <p>This operation does not produce sign or overflow conditions. Only the .e/z or .ne/.nz conditional modifiers should be used.</p> <p>Register source operands can use source modifiers: Any source modifier is logical, optionally changing a source value s to ~s (inverting all source bits). This capability allows expressions like a OR (NOT b) to be calculated with one instruction.</p>			
<p>Format: [(pred)] or[.cmod] (exec_size) dst src0 src1</p>			
Restriction			
Restriction : Source modifier is not allowed if source is an accumulator.			
Syntax			
[(pred)] or[.cmod] (exec_size) reg reg reg [(pred)] or[.cmod] (exec_size) reg reg imm32			
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	Saturation	Source Modifier
Y	Y	N	Y
Src Types		Dst Types	
*B,*W,*D		*B,*W,*D	
*W,*D		*W,*D	
*W,*D		*Q	
*Q		*W,*D	
*Q		*Q	
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG



or - Logic Or		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Logic Xor

xor - Logic Xor			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>The xor instruction performs component-wise logic XOR operation between src0 and src1 and stores the results in dst.</p> <p>This operation does not produce sign or overflow conditions. Only the .e/z or .ne/.nz conditional modifiers should be used.</p> <p>Register source operands can use source modifiers: Any source modifier is logical, optionally changing a source value s to ~s (inverting all source bits). This capability allows expressions like a XOR (NOT b) to be calculated with one instruction.</p>			
<p>Format: [(pred)] xor[.cmod] (exec_size) dst src0 src1</p>			
Restriction			
Restriction : Source modifier is not allowed if source is an accumulator.			
Syntax			
[(pred)] xor[.cmod] (exec_size) reg reg reg [(pred)] xor[.cmod] (exec_size) reg reg imm32			
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	Saturation	Source Modifier
Y	Y	N	Y
Src Types		Dst Types	
*B,*W,*D		*B,*W,*D	
*W,*D		*W,*D	
*W,*D		*Q	
*Q		*W,*D	
*Q		*Q	
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	((RegSource)[Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG



xor - Logic Xor		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
		Format: EU_INSTRUCTION_SOURCES_REG_IMM
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



MEDIA_CURBE_LOAD

MEDIA_CURBE_LOAD			
Source:	RenderCS		
Length Bias:	2		
Workaround : See "GPGPU Command Workarounds" section for additional programming constraints for this command.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	0h MEDIA_CURBE_LOAD
		Format:	OpCode
	23:16	SubOpcode	
Default Value:		1h MEDIA_CURBE_LOAD SubOp	
Format:		OpCode	
15:0	DWord Length		
	Format:	=n Total Length - 2	
	Value	Name	Description
	2h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)
1	31:0	Reserved	
		Format:	MBZ
2	31:17	Reserved	
		Format:	MBZ
	16:0	CURBE Total Data Length	
		Format:	U17 In Bytes
Description			
This field provides the length in bytes of the CURBE data. This field must have the same alignment as the Curbe Object Data Start Address.As the CURBE data are sent directly to ROB, range is limited to CURBE Allocation Size.			
This field must be 64-byte aligned.			



MEDIA_CURBE_LOAD						
3	31:0	CURBE Data Start Address				
		Format: DynamicStateOffset[31:0] CURBE				
		Description				
		Specifies the 64-byte aligned address of the CURBE data. This pointer is relative to the Dynamics Base Address .				
		<table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>[0, FFFFFFFh]</td><td></td></tr></tbody></table>	Value	Name	[0, FFFFFFFh]	
Value	Name					
[0, FFFFFFFh]						



MEDIA_INTERFACE_DESCRIPTOR_LOAD

MEDIA_INTERFACE_DESCRIPTOR_LOAD						
Source:	RenderCS					
Length Bias:	2					
<p>A Media_State_Flush should be used before this command to ensure that the temporary Interface Descriptor storage is cleared.</p> <p>Workaround : See "GPGPU Command Workarounds" section for additional programming constraints for this command.</p>						
DWord	Bit	Description				
0	31:29	Command Type				
		Default Value: 3h GFXPIPE				
		Format: OpCode				
	28:27	Pipeline				
		Default Value: 2h Media Format: OpCode				
26:24	Media Command Opcode					
	Default Value: 0h MEDIA_INTERFACE_DESCRIPTOR_LOAD Format: OpCode					
23:16	SubOpcode					
	Default Value: 2h MEDIA_INTERFACE_DESCRIPTOR_LOAD SubOp Format: OpCode					
15:0	DWord Length					
	Format: =n Total Length - 2					
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>2h</td> <td>DWORD_COUNT_n [Default]</td> <td>Excludes DWord (0,1)</td> </tr> </tbody> </table>	Value	Name	Description	2h	DWORD_COUNT_n [Default]
Value	Name	Description				
2h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)				
1	31:0	Reserved				
		Format: MBZ				
2	31:17	Reserved				
		Format: MBZ				
	16:0	Interface Descriptor Total Length				
		Format: U17 In bytes				
<p>This field provides the length in bytes of the Interface Descriptor data. This field must have the same alignment as the Interface Descriptor Data Start Address. It must be DQWord (32-byte) aligned. As the Interface Descriptor data are sent directly to ROB, range is limited to CURBE Allocation Size.</p>						



MEDIA_INTERFACE_DESCRIPTOR_LOAD													
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[32,2048]</td> <td>[1,64] interface descriptor entries</td> </tr> </tbody> </table>	Value	Name	[32,2048]	[1,64] interface descriptor entries								
Value	Name												
[32,2048]	[1,64] interface descriptor entries												
3	<table border="1"> <tr> <td style="text-align: center;">31:0</td> <td> <p>Interface Descriptor Data Start Address</p> <p>Format: DynamicStateOffset[31:0]INTERFACE_DESCRIPTOR_DATA</p> </td> </tr> <tr> <td colspan="2" style="text-align: center;">Description</td> </tr> <tr> <td colspan="2"> <p>This bit specifies the <u>64-byte</u> aligned address of the Interface Descriptor data. This pointer is relative to the Dynamics Base Address.</p> </td> </tr> <tr> <td colspan="2"> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, FFFFFFFFh]</td> <td></td> </tr> </tbody> </table> </td> </tr> </table>	31:0	<p>Interface Descriptor Data Start Address</p> <p>Format: DynamicStateOffset[31:0]INTERFACE_DESCRIPTOR_DATA</p>	Description		<p>This bit specifies the <u>64-byte</u> aligned address of the Interface Descriptor data. This pointer is relative to the Dynamics Base Address.</p>		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, FFFFFFFFh]</td> <td></td> </tr> </tbody> </table>		Value	Name	[0, FFFFFFFFh]	
31:0	<p>Interface Descriptor Data Start Address</p> <p>Format: DynamicStateOffset[31:0]INTERFACE_DESCRIPTOR_DATA</p>												
Description													
<p>This bit specifies the <u>64-byte</u> aligned address of the Interface Descriptor data. This pointer is relative to the Dynamics Base Address.</p>													
<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, FFFFFFFFh]</td> <td></td> </tr> </tbody> </table>		Value	Name	[0, FFFFFFFFh]									
Value	Name												
[0, FFFFFFFFh]													



MEDIA_OBJECT

MEDIA_OBJECT			
Source:		RenderCS	
Length Bias:		2	
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	OpCode
	28:27	Media Command Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	1h MEDIA_OBJECT
		Format:	OpCode
	23:16	Media Command Sub-Opcode	
Default Value:		0h MEDIA_OBJECT SubOp	
Format:		OpCode	
15:0	DWord Length	Default Value:	4h DWORD_COUNT_n
		Format:	=n Total Length - 2
		Excludes DWords 0,1 Generic Mode: DWord Length = N+4, where N is in the range of [0,504]. The maximum is 504 DW (equivalent to 63 8-DW registers). 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 N and indirect data length rounded up to 8-DW aligned individually). The minimal inline data length is 0.	
	31:8	Reserved	
		7:6	Reserved
			Format:
	5:0	Interface Descriptor Offset	
		Format:	U6
		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.	



MEDIA_OBJECT													
2	31	<p>Children Present</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Indicates that the root thread may send spawn messages to spawn child threads and/or synchronized root threads.</p> <p>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.</p> <p>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.</p> <p><i>In order avoid deadlock, such dereference must be issued once and only once for each URB handle.</i></p>	Format:	Enable									
	Format:	Enable											
	30:27	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ											
	26:25	<p>Reserved</p>											
	24	<p>Thread Synchronization</p> <p>This field when set indicates that the dispatch of the thread originated from this command is based on the "spawn root thread" message.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>No thread synchronization</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Thread dispatch is synchronized by the 'spawn root thread' message</td> </tr> </tbody> </table>	Value	Name	0	No thread synchronization	1	Thread dispatch is synchronized by the 'spawn root thread' message					
	Value	Name											
	0	No thread synchronization											
	1	Thread dispatch is synchronized by the 'spawn root thread' message											
	23	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												
22	<p>Force Destination</p> <p>If set, bits 20:17 are used to determine the destination of this dispatch, if clear the destination will be chosen based on load.</p>												
21	<p>Use Scoreboard</p> <p>This field specifies whether the thread associated with this command uses hardware scoreboard. Only when this field is set, the scoreboard control fields in the VFE Dword are valid. If this field is cleared, the thread associated with this command bypasses hardware scoreboard.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Not using scoreboard</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Using scoreboard</td> </tr> </tbody> </table>	Value	Name	0	Not using scoreboard	1	Using scoreboard						
Value	Name												
0	Not using scoreboard												
1	Using scoreboard												
20:19	<p>Slice Destination Select</p> <p>This bit along with the subslice destination select determines the slice that this thread must be sent to. Ignored if Force Destination = 0, or if product only has 1 slice.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Slice 0</td> <td></td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Slice 1</td> <td>Cannot be used in products without a Slice 1.</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Slice 2</td> <td>Cannot be used in products without a Slice 2.</td> </tr> </tbody> </table>	Value	Name	Description	00b	Slice 0		01b	Slice 1	Cannot be used in products without a Slice 1.	10b	Slice 2	Cannot be used in products without a Slice 2.
Value	Name	Description											
00b	Slice 0												
01b	Slice 1	Cannot be used in products without a Slice 1.											
10b	Slice 2	Cannot be used in products without a Slice 2.											



MEDIA_OBJECT												
	11b	Reserved										
	18:17	<p>SubSlice Destination Select</p> <p>This field selects the SubSlice that this thread must be sent to. Ignored if Force Destination = 0</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>11b</td> <td>Subslice 3</td> </tr> <tr> <td>10b</td> <td>SubSlice 2</td> </tr> <tr> <td>01b</td> <td>SubSlice 1</td> </tr> <tr> <td>00b</td> <td>SubSlice 0</td> </tr> </tbody> </table>	Value	Name	11b	Subslice 3	10b	SubSlice 2	01b	SubSlice 1	00b	SubSlice 0
	Value	Name										
	11b	Subslice 3										
	10b	SubSlice 2										
	01b	SubSlice 1										
00b	SubSlice 0											
16:0	<p>Indirect Data Length</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>U17 In bytes</td> </tr> </table> <p>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 (URB Entry Allocation Size)*(Number of URB Entries) in the MEDIA_VFE_STATE command. When both inline and indirect data are fetched for this command, enough room should be allowed for at least 1 copy of the indirect and 1 copy of the inline data in the URB.</p>	Format:	U17 In bytes									
Format:	U17 In bytes											
3	31:0	<p>Indirect Data Start Address</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>GraphicsAddress[31:0]</td> </tr> </table>	Format:	GraphicsAddress[31:0]								
		Format:	GraphicsAddress[31:0]									
		Description										
		<p>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 on the mode of operation.</p> <p>This field specifies the 64-byte aligned address of the indirect data.</p>										
		Value										
		Name										
[0,512MB]												
Programming Notes												
Bits 31:29 MBZ												
4	31:25	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
24:16	<p>Scoreboard Y</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>U9</td> </tr> </table> <p>This field provides the Y term of the scoreboard value of the current thread.</p>	Format:	U9									
Format:	U9											



MEDIA_OBJECT		
	15:9	Reserved Format: MBZ
	8:0	Scoreboard X Format: U9 This field provides the X term of the scoreboard value of the current thread.
5	31:20	Reserved Format: MBZ
	19:16	Scoreboard Color Format: U4 This field specifies which dependency color the current thread belongs to. It affects the dependency scoreboard control.
	15:8	Reserved Format: MBZ
	7:0	Scoreboard Mask Format: Boolean Each bit indicates the corresponding dependency scoreboard is dependent on. This field is AND'd with the corresponding Scoreboard Mask field in the MEDIA_VFE_STATE command. Bit n (for n = 0...7): Scoreboard n is dependent, where bit 0 maps to n = 0.
6..n	31:0	Inline Data Generic Mode: The format of this data is specified by software. Hardware does not interpret this data; it merely passes it to the kernel for processing. The total size for the inline data and indirect data must not exceed 112 registers.



MEDIA_OBJECT_GRPID

MEDIA_OBJECT_GRPID			
Source:	RenderCS		
Length Bias:	2		
<p>The MEDIA_OBJECT_GRPID command is a variation of MEDIA_OBJECT which includes a group id which is used to allocate and track Barriers and Shared Local Memory. The Interface Descriptor is used to specify how much SLM is needed and how many threads will be reporting to the Barrier. All MEDIA_OBJECT_GRPIDs with the same group id should have the same interface descriptor and be dispatched to the same Tslice – the dispatcher will ensure this if Force Destination = 0, but software must ensure this if Force Destination = 1. Software should also ensure that all the threads needed for the Barrier will fit into a Tslice, or the Barrier will never be satisfied. Either SLM or a barrier must be used with MEDIA_OBJECT_GRPID, if neither is needed then a MEDIA_OBJECT must be used instead.</p> <p>MEDIA_OBJECT_GRPID supports the GPGPU version of payload delivery – either indirect or CURBE can be split between the threads in a group (per-thread payload), as well as a section which is sent to all threads (cross-thread payload). See the GPGPU payload section. For indirect, the same pointer must be sent with all the commands associated with the thread group for payload splitting to work properly. Inline data is not split, but the payload attached to each command is sent with that thread. Only one of inline, indirect, or CURBE is allowed, but at least one form of payload must be sent.</p> <p>MEDIA_STATE_FLUSH with the watermark bit must be placed between groups created by MEDIA_OBJECT_GRPID. The Interface Descriptor associated with the watermark must match the Interface Descriptor used for the following group.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	OpCode
	28:27	Media Command Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	1h MEDIA_OBJECT_GRPID
		Format:	OpCode
	23:16	Media Command Sub-Opcode	
		Default Value:	6h MEDIA_OBJECT_GRPID SubOp
		Format:	OpCode
	15:0	DWord Length	
		Default Value:	5h DWORD_COUNT_n
		Format:	=n Total Length - 2
Excludes DWords 0,1			
Generic Mode: DWord Length = N+5, where N is in the range of [0,504]. The maximum is 504			



MEDIA_OBJECT_GRPID																
		<p>DW (equivalent to 63 8-DW registers). 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 N and indirect data length rounded up to 8-DW aligned individually). The minimal inline data length is 0.</p>														
1	31:8	Reserved														
	7:6	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
	5:0	<p>Interface Descriptor Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U6</td> </tr> </table> <p>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.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,30]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,30]									
Format:	U6															
Value	Name															
[0,30]																
2	31:25	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
	24	Reserved														
	23	<p>End of Thread Group This bit indicates that this dispatch is the last for the current thread group.</p>														
	22	<p>Force Destination If set, bits 20:17 are used to determine the destination of this dispatch, if clear the destination will be chosen based on load.</p>														
	21	<p>Use Scoreboard This field specifies whether the thread associated with this command uses hardware scoreboard. Only when this field is set, the scoreboard control fields in the VFE Dword are valid. If this field is cleared, the thread associated with this command bypasses hardware scoreboard.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Not using scoreboard</td> </tr> <tr> <td>1</td> <td>Using scoreboard</td> </tr> </tbody> </table>	Value	Name	0	Not using scoreboard	1	Using scoreboard								
	Value	Name														
	0	Not using scoreboard														
1	Using scoreboard															
20:19	<p>Slice Destination Select This bit along with the SubSlice destination select determines the slice that this thread must be sent to. Ignored if Force Destination = 0, or if product only has 1 slice.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Slice 0</td> <td></td> </tr> <tr> <td>01b</td> <td>Slice 1</td> <td>Cannot be used in products without a Slice 1.</td> </tr> <tr> <td>10b</td> <td>Slice 2</td> <td>Cannot be used in products without a Slice 2.</td> </tr> <tr> <td>11b</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	00b	Slice 0		01b	Slice 1	Cannot be used in products without a Slice 1.	10b	Slice 2	Cannot be used in products without a Slice 2.	11b	Reserved	
Value	Name	Description														
00b	Slice 0															
01b	Slice 1	Cannot be used in products without a Slice 1.														
10b	Slice 2	Cannot be used in products without a Slice 2.														
11b	Reserved															



MEDIA_OBJECT_GRPID												
	18:17	<p>SubSlice Destination Select</p> <p>This field selects the SubSlice that this thread must be sent to. Ignored if Force Destination = 0</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>11b</td> <td>Subslice3</td> </tr> <tr> <td>10b</td> <td>SubSlice 2</td> </tr> <tr> <td>01b</td> <td>SubSlice 1</td> </tr> <tr> <td>00b</td> <td>SubSlice 0</td> </tr> </tbody> </table>	Value	Name	11b	Subslice3	10b	SubSlice 2	01b	SubSlice 1	00b	SubSlice 0
	Value	Name										
11b	Subslice3											
10b	SubSlice 2											
01b	SubSlice 1											
00b	SubSlice 0											
	16:0	<p>Indirect Data Length</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>U17 In bytes</td> </tr> </table> <p>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 (URB Entry Allocation Size)*(Number of URB Entries) in the MEDIA_VFE_STATE command.</p>	Format:	U17 In bytes								
Format:	U17 In bytes											
3	31:0	<p>Indirect Data Start Address</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>GraphicsAddress[31:0]</td> </tr> </table> <p style="text-align: center;">Description</p> <p>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 on the mode of operation.</p> <p>It is the 64-byte aligned address of the indirect data.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Value</th> <th style="width: 33%;">Name</th> <th style="width: 33%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0-512MB]</td> <td></td> <td>Bits 31:29 MBZ</td> </tr> </tbody> </table>	Format:	GraphicsAddress[31:0]	Value	Name	Description	[0-512MB]		Bits 31:29 MBZ		
Format:	GraphicsAddress[31:0]											
Value	Name	Description										
[0-512MB]		Bits 31:29 MBZ										
4	31:25	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	24:16	<p>Scoreboard Y</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>U9</td> </tr> </table> <p>This field provides the Y term of the scoreboard value of the current thread.</p>	Format:	U9								
	Format:	U9										
15:9	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
8:0	<p>Scoreboard X</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>U9</td> </tr> </table> <p>This field provides the X term of the scoreboard value of the current thread.</p>	Format:	U9									
Format:	U9											



MEDIA_OBJECT_GRPID		
5	31:20	Reserved Format: MBZ
	19:16	Scoreboard Color Format: U4 This field specifies which dependency color the current thread belongs to. It affects the dependency scoreboard control.
	15:8	Reserved Format: MBZ
	7:0	Scoreboard Mask Format: Boolean Each bit indicates the corresponding dependency scoreboard is dependent on. This field is AND'd with the corresponding Scoreboard Mask field in the MEDIA_VFE_STATE command. Bit n (for n = 0...7): Scoreboard n is dependent, where bit 0 maps to n = 0.
6	31:0	GroupID A unique identifying number which describes the threads which share a barrier and/or SLM. Reuse of numbers is allowed as long as the old group is not currently running.
7..n	31:0	Inline Data The format of this data is specified by software. Hardware does not interpret this data; it merely passes it to the kernel for processing. The total size for the inline data and indirect data must not exceed 112 registers.



MEDIA_OBJECT_PRT

MEDIA_OBJECT_PRT						
Source:	RenderCS					
Length Bias:	2					
<p>command is for generating Persistent Root Thread for the media pipeline. It only supports loading of inline data but not indirect data.</p> <p>This command should be used for a root thread that might have to be present in the system for a period longer than the certain minimal context-switch interrupt latency. It has to honor the context interrupt signal to terminate upon request. It should also handle replay from the interrupted point upon context restore (the same thread being dispatched more than once). In contrary, if a thread is not a Persistent Root Thread, if dispatched, it must run to completion.</p> <p>The command can be used in all VFE modes, except VLD mode.</p> <p>For simplification, _PRT command has a fixed size of 16 DWORD</p>						
DWord	Bit	Description				
0	31:29	Command Type				
		Default Value:	3h GFXPIPE			
		Format:	OpCode			
	28:27	Pipeline				
		Default Value:	2h Media			
		Format:	OpCode			
	26:24	Media Command Opcode				
		Default Value:	1h MEDIA_OBJECT_PRT			
		Format:	OpCode			
	23:16	SubOpcode				
Default Value:		2h MEDIA_OBJECT_PRT SubOp				
Format:		OpCode				
15:0	DWord Length					
	Format:	=n Total Length - 2				
	Note: Regardless of the mode, inline data must be present in this command. The command size must fit within 16 dwords.					
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0Eh</td> <td>DWORD_COUNT_n [Default]</td> <td>Excludes DWord (0,1)</td> </tr> </tbody> </table>	Value	Name	Description	0Eh	DWORD_COUNT_n [Default]
Value	Name	Description				
0Eh	DWORD_COUNT_n [Default]	Excludes DWord (0,1)				
1	31:6	Reserved				
		Format:	MBZ			
	5:0	Interface Descriptor Offset				
Format:	U6					
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.						



MEDIA_OBJECT_PRT										
2	31	<p>Children Present</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>Indicates that the root thread may send spawn messages to spawn child threads and/or synchronized root threads.</p> <p>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.</p> <p>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.</p> <p>In order avoid deadlock, such de-reference must be issued once and only once for each URB handle.</p>	Format:	Enable						
	Format:	Enable								
	30:24	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
	Format:	MBZ								
	23	<p>PRT_Fence Needed</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field specifies that a PRT_Fence is generated after dispatching the thread associated with this 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.</p>	Format:	Enable						
Format:	Enable									
22	<p>PRT_FenceType</p> <p>This field specifies the type of fence the PRT thread uses. If this field is set to 0, the fence is set at the end of the root thread queue. It will block the dispatch of the next root thread, but allowed these root threads to be populated through VFE to the root thread queue in TS. If this field is set to 1, the fence is set at the entry of VFE, similar to the fence set by the MEDIA_STATE_FLUSH command. No more command can go into the media pipe until a thread spawn message is sent (by the PRT).</p> <p>This field is only valid when PRT_Fence Needed is set to 1. Otherwise, it is ignored by hardware.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Root thread queue</td> <td>Root thread queue fence</td> </tr> <tr> <td>1h</td> <td>VFE state flush</td> <td>VFE state flush fence</td> </tr> </tbody> </table>	Value	Name	Description	0h	Root thread queue	Root thread queue fence	1h	VFE state flush	VFE state flush fence
Value	Name	Description								
0h	Root thread queue	Root thread queue fence								
1h	VFE state flush	VFE state flush fence								
21:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
3	31:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ									
4..15	31:0	<p>Inline Data</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U32</td> </tr> </table>	Format:	U32						
Format:	U32									



MEDIA_OBJECT_WALKER

MEDIA_OBJECT_WALKER			
Source:		RenderCS	
Length Bias:		2	
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	1h MEDIA_OBJECT_WALKER
		Format:	OpCode
	23:16	SubOpcode	
		Default Value:	03h MEDIA_OBJECT_WALKER SubOp
		Format:	OpCode
	15:0	DWord Length	
		Default Value:	0Fh DWORD_COUNT_n
		Format:	=n Total Length - 2
<p>Note: If this field is greater than 15, it indicates that inline data is present. If present, inline data is common for all threads generated from this command, If this field is 15, it indicates that inline data is not present. It should be noted that unlike other media object command, inline data is optional for this command.</p>			
1	31:8	Reserved	
	7:6	Reserved	
		Format:	Reserved
5:0	Interface Descriptor Offset		
Format:		U6	
<p>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.</p>			
2	31:25	Reserved	
		Format:	MBZ
	24	Thread Synchronization	
<p>This field when set indicates that the dispatch of the thread originated from this command is based on the "spawn root thread" message.</p>			



MEDIA_OBJECT_WALKER																			
		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>No thread synchronization</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Thread dispatch is synchronized by the 'spawn root thread' message</td> </tr> </tbody> </table>	Value	Name	0	No thread synchronization	1	Thread dispatch is synchronized by the 'spawn root thread' message											
Value	Name																		
0	No thread synchronization																		
1	Thread dispatch is synchronized by the 'spawn root thread' message																		
	23:22	<p>Masked Dispatch</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Enable the masking of the dispatch of individual threads based on a bitmask read from CURBE, and specifies the pitch of the CURBE surface. If enabled, CURBE will not be used for thread payload.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td></td> <td>Masked Dispatch Disabled</td> </tr> <tr> <td style="text-align: center;">01b</td> <td></td> <td>Masked Dispatch with 128-bit pitch in CURBE</td> </tr> <tr> <td style="text-align: center;">10b</td> <td></td> <td>Masked Dispatch with 256-bit pitch in CURBE</td> </tr> <tr> <td style="text-align: center;">11b</td> <td></td> <td>Masked Dispatch with 512-bit pitch in CURBE</td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	00b		Masked Dispatch Disabled	01b		Masked Dispatch with 128-bit pitch in CURBE	10b		Masked Dispatch with 256-bit pitch in CURBE	11b		Masked Dispatch with 512-bit pitch in CURBE
Format:	U2																		
Value	Name	Description																	
00b		Masked Dispatch Disabled																	
01b		Masked Dispatch with 128-bit pitch in CURBE																	
10b		Masked Dispatch with 256-bit pitch in CURBE																	
11b		Masked Dispatch with 512-bit pitch in CURBE																	
	21	<p>Use Scoreboard</p> <p>This field specifies whether the thread associated with this command uses hardware scoreboard. Only when this field is set, the scoreboard control fields in the VFE Dword are valid. If this field is cleared, the thread associated with this command bypasses hardware scoreboard.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Not using scoreboard</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Using scoreboard</td> </tr> </tbody> </table>	Value	Name	0	Not using scoreboard	1	Using scoreboard											
Value	Name																		
0	Not using scoreboard																		
1	Using scoreboard																		
	20:17	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																		
	16:0	<p>Indirect Data Length</p> <table border="1"> <tr> <td>Format:</td> <td>U17 in bytes</td> </tr> </table> <p>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, the range is limited to the (URB Entry Allocation Size)*(Number of URB Entries) specified in the MEDIA_VFE_STATE command.</p>	Format:	U17 in bytes															
Format:	U17 in bytes																		
3	31:0	<p>Indirect Data Start Address</p> <p>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 on the mode of operation.</p> <p>It is the 64-byte aligned address of the indirect data</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0 - 512MB]</td> <td></td> <td>(Bits 31:29 MBZ)</td> </tr> </tbody> </table>	Value	Name	Description	[0 - 512MB]		(Bits 31:29 MBZ)											
Value	Name	Description																	
[0 - 512MB]		(Bits 31:29 MBZ)																	



MEDIA_OBJECT_WALKER																							
4	31:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ																			
Format:	MBZ																						
5	31:8	<p>Group ID Loop Select</p> <p>This bit field chooses which of the nested loops of the walker are used to identify threads which share a group id and therefore a shared barrier and SLM. The programmer must ensure that each group will fit into a single subslice. When barriers are enabled every group must have the same number of threads matching the number specified in the Interface Descriptor.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>No_Groups</td> <td>Groups are not created, barriers and SLM are not allocated</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Color_Groups</td> <td>Each complete iteration of the Color loop defines a group, the group id is the concatenation of the Outer global, Inner global, Outer local, Mid local and Inner local loop execution counts.</td> </tr> <tr> <td style="text-align: center;">2</td> <td>InnerLocal_Groups</td> <td>Each complete iteration of the Inner local loop and Color loop defines a group, the group id is the concatenation of the Outer global loop to the Mid local loop execution counts.</td> </tr> <tr> <td style="text-align: center;">3</td> <td>MidLocal_Groups</td> <td>Each complete iteration of the Mid local loop and lower loops defines a group, the group id is the concatenation of the Outer global loop to the Outer local loop execution counts.</td> </tr> <tr> <td style="text-align: center;">4</td> <td>OuterLocal_Groups</td> <td>Each complete iteration of the Outer local loop and lower loops defines a group, the group id is the concatenation of the Outer global loop and the Inner global loop execution counts.</td> </tr> <tr> <td style="text-align: center;">5</td> <td>InnerGlobal_Groups</td> <td>Each complete iteration of the Inner global loop and lower loops defines a group, the group id is the Outer global loop execution count.</td> </tr> </tbody> </table>	Value	Name	Description	0	No_Groups	Groups are not created, barriers and SLM are not allocated	1	Color_Groups	Each complete iteration of the Color loop defines a group, the group id is the concatenation of the Outer global, Inner global, Outer local, Mid local and Inner local loop execution counts.	2	InnerLocal_Groups	Each complete iteration of the Inner local loop and Color loop defines a group, the group id is the concatenation of the Outer global loop to the Mid local loop execution counts.	3	MidLocal_Groups	Each complete iteration of the Mid local loop and lower loops defines a group, the group id is the concatenation of the Outer global loop to the Outer local loop execution counts.	4	OuterLocal_Groups	Each complete iteration of the Outer local loop and lower loops defines a group, the group id is the concatenation of the Outer global loop and the Inner global loop execution counts.	5	InnerGlobal_Groups	Each complete iteration of the Inner global loop and lower loops defines a group, the group id is the Outer global loop execution count.
Value	Name	Description																					
0	No_Groups	Groups are not created, barriers and SLM are not allocated																					
1	Color_Groups	Each complete iteration of the Color loop defines a group, the group id is the concatenation of the Outer global, Inner global, Outer local, Mid local and Inner local loop execution counts.																					
2	InnerLocal_Groups	Each complete iteration of the Inner local loop and Color loop defines a group, the group id is the concatenation of the Outer global loop to the Mid local loop execution counts.																					
3	MidLocal_Groups	Each complete iteration of the Mid local loop and lower loops defines a group, the group id is the concatenation of the Outer global loop to the Outer local loop execution counts.																					
4	OuterLocal_Groups	Each complete iteration of the Outer local loop and lower loops defines a group, the group id is the concatenation of the Outer global loop and the Inner global loop execution counts.																					
5	InnerGlobal_Groups	Each complete iteration of the Inner global loop and lower loops defines a group, the group id is the Outer global loop execution count.																					
	7:0	<p>Scoreboard Mask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">Boolean</td> </tr> </table> <p>Each bit indicates the corresponding dependency scoreboard is dependent on. This field is AND'd with the corresponding Scoreboard Mask field in the MEDIA_VFE_STATE. All threads generated by this walker command share the same dynamic mask.</p> <p>Bit n (for n = 0...7): Scoreboard n is dependent, where bit 0 maps to n = 0.</p>	Format:	Boolean																			
Format:	Boolean																						
6	31:28	<p>Reserved</p>																					
	27:24	<p>Color Count Minus One</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U4</td> </tr> </table> <p>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.</p> <p>Usage Example: This allows multiple sets of dependency threads to be dispatched.</p>	Format:	U4																			
Format:	U4																						
	23:21	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ																			
Format:	MBZ																						



MEDIA_OBJECT_WALKER		
	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
	9:8	Mid-Loop Unit X Format: S1
	7:0	Reserved Format: MBZ
7	31:28	Reserved Format: MBZ
	27:16	Global Loop Exec Count Format: U12
	15:12	Reserved Format: MBZ
	11:0	Local Loop Exec Count Format: U12
8	31:27	Reserved Format: MBZ
	26:16	Block Resolution Y Format: U11 Vertical resolution of the local loop.
	15:11	Reserved Format: MBZ
	10:0	Block Resolution X Format: U11 Horizontal resolution of the local loop.
9	31:27	Reserved Format: MBZ
	26:16	Local Start Y Format: U11 Starting vertical position of the local loop.



MEDIA_OBJECT_WALKER		
	15:11	Reserved Format: MBZ
	10:0	Local Start X Format: U11 Starting horizontal position of the local loop.
10	31:0	Reserved Format: MBZ
11	31:28	Reserved Format: MBZ
	27:16	Local Outer Loop Stride Y Format: S11 Vertical stride of the local outer loop, in 2's complement.
	15:12	Reserved Format: MBZ
	11:0	Local Outer Loop Stride X Format: S11 Horizontal stride of the local outer loop, in 2's complement.
12	31:28	Reserved Format: MBZ
	27:16	Local Inner Loop Unit Y Format: S11 Vertical stride of the local inner loop, in 2's complement.
	15:12	Reserved Format: MBZ
	11:0	Local Inner Loop Unit X Format: S11 Horizontal stride of the local inner loop, in 2's complement.
13	31:27	Reserved Format: MBZ
	26:16	Global Resolution Y Format: U11 Vertical resolution of the global loop.



MEDIA_OBJECT_WALKER		
	15:11	Reserved Format: _____ MBZ
	10:0	Global Resolution X Format: _____ U11 Horizontal resolution of the global loop.
14	31:28	Reserved Format: _____ MBZ
	27:16	Global Start Y Format: _____ S11 Starting vertical location of the global loop, in 2's complement.
	15:12	Reserved Format: _____ MBZ
	11:0	Global Start X Format: _____ S11 Starting horizontal location of the global loop, in 2's complement.
15	31:28	Reserved Format: _____ MBZ
	27:16	Global Outer Loop Stride Y Format: _____ S11 Vertical stride of the global outer loop, in 2's complement.
	15:12	Reserved Format: _____ MBZ
	11:0	Global Outer Loop Stride X Format: _____ S11 Horizontal stride of the global outer loop, in 2's complement.
16	31:28	Reserved Format: _____ MBZ
	27:16	Global Inner Loop Unit Y Format: _____ S11 Vertical stride of the global inner loop, in 2's complement.
	15:12	Reserved Format: _____ MBZ



MEDIA_OBJECT_WALKER						
	11:0	Global Inner Loop Unit X <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S11</td> </tr> <tr> <td colspan="2">Horizontal stride of the global inner loop, in 2's complement.</td> </tr> </table>	Format:	S11	Horizontal stride of the global inner loop, in 2's complement.	
Format:	S11					
Horizontal stride of the global inner loop, in 2's complement.						
17..n	31:0	Inline Data				



MEDIA_STATE_FLUSH

MEDIA_STATE_FLUSH						
Source:	RenderCS					
Length Bias:	2					
<p>This command updates the Message Gateway state. In particular, it updates the state for a selected Interface Descriptor.</p> <p>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.</p> <p>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:</p> <pre>MEDIA_STATE_FLUSH VFE_STATE // VFE will hold CS if watermark isn't met MI_ARB_OFF // There must be at least one VFE command before this one MEDIA_OBJECT ... MI_ARB_ON</pre>						
DWord	Bit	Description				
0	31:29	Command Type				
		Default Value: 3h GFXPIPE				
	Format: OpCode					
	28:27	Pipeline				
		Default Value: 2h Media				
Format: OpCode						
26:24	Media Command Opcode					
	Default Value: 0h MEDIA_STATE_FLUSH					
Format: OpCode						
23:16	SubOpcode					
	Default Value: 4h MEDIA_STATE_FLUSH SubOp					
Format: OpCode						
15:0	DWord Length					
	Format: =n Total Length - 2					
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>DWORD_COUNT_n [Default]</td> <td>Excludes DWord (0,1)</td> </tr> </tbody> </table>	Value	Name	Description	0h	DWORD_COUNT_n [Default]
Value	Name	Description				
0h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)				
1	31:9	Reserved				
Format: MBZ						



MEDIA_STATE_FLUSH			
8	Reserved		
7	<p>Flush to GO</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>This bit indicates that the write data out of this thread group should be flushed to the point where it is visible to following commands.</p>	Format:	Enable
Format:	Enable		
6	<p>Watermark Required</p> <p>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.</p> <p>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 descriptor requires that this bit is clear to ensure the ID cache is reloaded.</p> <p>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.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>The Interface Descriptor Offset used for the flush must be the same as that used for the GPGPU_OBJECTs. GPGPU_WALKER automatically checks the Watermark conditions before starting a thread, so this bit should not be set before GPGPU_WALKER.</p>	Programming Notes	
Programming Notes			
5:0	<p>Interface Descriptor Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6
Format:	U6		



MEDIA_VFE_STATE

MEDIA_VFE_STATE								
Source:	RenderCS							
Length Bias:	2							
<p>A stalling PIPE_CONTROL is required before MEDIA_VFE_STATE unless the only bits that are changed are scoreboard related: Scoreboard Enable, Scoreboard Type, Scoreboard Mask, Scoreboard * Delta. For these scoreboard related states, a MEDIA_STATE_FLUSH is sufficient.</p> <ul style="list-style-type: none"> • MEDIA_STATE_FLUSH (optional, only if barrier dependency is needed) • MEDIA_INTERFACE_DESCRIPTOR_LOAD (optional) 								
DWord	Bit	Description						
0	31:29	Command Type						
		Default Value:	3h GFXPIPE					
		Format:	OpCode					
	28:27	Pipeline						
		Default Value:	2h Media					
	26:24	Media Command Opcode						
		Default Value:	0h MEDIA_VFE_STATE					
	23:16	SubOpcode						
		Default Value:	0h MEDIA_VFE_STATE SubOp					
	15:0	15:0	DWord Length					
Format:			=n Total Length - 2					
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>07h</td> <td>DWORD_COUNT_n [Default]</td> <td>Excludes DWord (0,1)</td> </tr> </tbody> </table>		Value	Name	Description	07h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)	
Value		Name	Description					
07h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)						
1	31:10	Scratch Space Base Pointer						
		Format:	GeneralStateOffset[31:10]					
	<p>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.</p>							
9:8	Reserved							
		Format:	MBZ					



MEDIA_VFE_STATE										
7:4	Stack Size	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td></td> <td>indicating [1KBytes, 2MBytes]</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Since the stack uses the upper portion of the scratch space, Stack Size = < Per Thread Scratch Space</p>	Value	Name	Description	[0,11]		indicating [1KBytes, 2MBytes]		
	Value	Name	Description							
[0,11]		indicating [1KBytes, 2MBytes]								
3:0	Per Thread Scratch Space	<table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>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 threads in the device each get Per Thread Scratch Space size without exceeding the driver-allocated scratch space.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td></td> <td>indicating [1k bytes, 2 Mbytes]: 0=1k, 1=2k, 2=4k, 3=8k ... 11=2M</td> </tr> </tbody> </table>	Format:	U4	Value	Name	Description	[0,11]		indicating [1k bytes, 2 Mbytes]: 0=1k, 1=2k, 2=4k, 3=8k ... 11=2M
Format:	U4									
Value	Name	Description								
[0,11]		indicating [1k bytes, 2 Mbytes]: 0=1k, 1=2k, 2=4k, 3=8k ... 11=2M								
2	31:16	Reserved								
	15:0	Scratch Space Base Pointer High								
3	31:16	Maximum Number of Threads								
	15:8	Number of URB Entries								
	31:16	Maximum Number of Threads								



MEDIA_VFE_STATE													
		Programming Notes											
		Please note that 0 is not allowed for this field.											
	7	Reset Gateway Timer This field controls the reset of the timestamp counter maintained in Message Gateway. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Maintaining the existing timestamp state</td> </tr> <tr> <td>1h</td> <td>Resetting relative timer and latching the global timestamp</td> </tr> </tbody> </table>	Value	Name	0h	Maintaining the existing timestamp state	1h	Resetting relative timer and latching the global timestamp					
Value	Name												
0h	Maintaining the existing timestamp state												
1h	Resetting relative timer and latching the global timestamp												
	6	Bypass Gateway Control This field configures Gateway to use a simple message protocol. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Maintaining OpenGateway/ForwardMsg/CloseGateway protocol (legacy mode)</td> </tr> <tr> <td>1h</td> <td>Bypassing OpenGateway/CloseGateway protocol</td> </tr> </tbody> </table>	Value	Name	0h	Maintaining OpenGateway/ForwardMsg/CloseGateway protocol (legacy mode)	1h	Bypassing OpenGateway/CloseGateway protocol					
Value	Name												
0h	Maintaining OpenGateway/ForwardMsg/CloseGateway protocol (legacy mode)												
1h	Bypassing OpenGateway/CloseGateway protocol												
	6	Reserved Format: <table border="1" style="width: 100%;"><tr><td style="width: 60%;"></td><td style="width: 40%;">MBZ</td></tr></table>		MBZ									
	MBZ												
	5:3	Reserved Format: <table border="1" style="width: 100%;"><tr><td style="width: 60%;"></td><td style="width: 40%;">MBZ</td></tr></table>		MBZ									
	MBZ												
	2	Reserved Format: <table border="1" style="width: 100%;"><tr><td style="width: 60%;"></td><td style="width: 40%;">MBZ</td></tr></table>		MBZ									
	MBZ												
	1:0	Reserved											
4	31:8	Reserved											
	7:4	Reserved Format: <table border="1" style="width: 100%;"><tr><td style="width: 60%;"></td><td style="width: 40%;">MBZ</td></tr></table>		MBZ									
		MBZ											
	3:2	Reserved Format: <table border="1" style="width: 100%;"><tr><td style="width: 60%;"></td><td style="width: 40%;">MBZ</td></tr></table>		MBZ									
	MBZ												
1:0	Slice Disable This field disables dispatch to slices and subslices for Media and GPGPU applications. It is used to limit the amount of scratch space that needs to be allocated for a context. If a particular configuration doesn't have slice or subslice then there is no impact to disabling it. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 55%;">Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>All Subslices Enabled</td> <td>All subslices are enabled.</td> </tr> <tr> <td>01b</td> <td>Only Slice 0 Enabled</td> <td>Slice 2 and 1 are disabled, only Slice 0 with all subslices is enabled.</td> </tr> <tr> <td>11b</td> <td>Only Slice 0 Subslice 0 Enabled</td> <td>Slice 2 and 1 are disabled, only Slice 0 with only subslice 0 enabled.</td> </tr> </tbody> </table>	Value	Name	Description	00b	All Subslices Enabled	All subslices are enabled.	01b	Only Slice 0 Enabled	Slice 2 and 1 are disabled, only Slice 0 with all subslices is enabled.	11b	Only Slice 0 Subslice 0 Enabled	Slice 2 and 1 are disabled, only Slice 0 with only subslice 0 enabled.
Value	Name	Description											
00b	All Subslices Enabled	All subslices are enabled.											
01b	Only Slice 0 Enabled	Slice 2 and 1 are disabled, only Slice 0 with all subslices is enabled.											
11b	Only Slice 0 Subslice 0 Enabled	Slice 2 and 1 are disabled, only Slice 0 with only subslice 0 enabled.											



		MEDIA_VFE_STATE	
5	31:16	URB Entry Allocation Size	
		Format:	U16-1
Description			
<p>Specifies the length of each URB entry used by the unit, in 256-bit register increments. ROB address for URB starts after CURBE Allocated region. (URB Entry Allocation Size * Number of URB Entries) + CURBE Allocation Size + Number of Interface Descriptors) must be less than (number of bytes allocated for the URB in L3CNTLREG / 32 bytes per entry). Note: Number of Interface Descriptors is 64.</p> <p>If SLM is enabled for GPGPU work then the number of available entries will be 1/2 the maximum URB entries.</p>			
Programming Notes			
<p>When Inline data is used with MEDIA_OBJECT or MEDIA_OBJECT_WALKER, then the URB entry allocation size must match the Inline data size. If Indirect data is being used with MEDIA_OBJECT or GPGPU_WALKER then the allocation size must be sufficient for the Indirect data. If both Inline and Indirect are being used, then the allocation size must match the sum of the Inline and Indirect.</p>			
5	15:0	CURBE Allocation Size	
		Format:	U16-1
Description			
<p>Specifies the total length allocated for CURBE, in 256-bit register increments. ROB address for CURBE starts at address 64. (URB Entry Allocation Size * Number of URB Entries) + CURBE Allocation Size + Interface Descriptor Entries) must be less than or equal to the number of entries in the URB as described in Configurations. Interface Descriptor Entries is 64</p> <p>If SLM is enabled for GPGPU work then the number of available entries will be ½ the maximum URB entries.</p>			
Programming Notes			
CURBE Allocation Size should be 0 for GPGPU workloads that uses indirect instead of CURBE.			
6	31	Scoreboard Enable	
		Format:	Enable
<p>This field enables and disables the hardware scoreboard in the Media Pipeline. If this field is cleared, hardware ignores the following scoreboard state fields.</p>			
		Value	Name
		0h	Scoreboard disabled
		1h	Scoreboard enabled



MEDIA_VFE_STATE							
30	<p>Scoreboard Type This field selects the type of scoreboard in use.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>Stalling Scoreboard</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>Non-Stalling Scoreboard</td> </tr> </tbody> </table>	Value	Name	0h	Stalling Scoreboard	1h	Non-Stalling Scoreboard
	Value	Name					
	0h	Stalling Scoreboard					
	1h	Non-Stalling Scoreboard					
<p>29:16 Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ						
<p>15:8 Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ						
<p>7:0 Scoreboard Mask</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable(8)</td> </tr> </table> <p>Each bit indicates the corresponding dependency scoreboard is enabled. The scoreboard is based on the relative (X, Y) distance from the current threads' (X, Y) position. Bit n (for n = 0...7): Score n is enabled.</p>	Format:	Enable(8)					
Format:	Enable(8)						
7	<p>31:28 Scoreboard 3 Delta Y</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>S3</td> </tr> </table> <p>Relative vertical distance of the dependent instance assigned to scoreboard 3, in the form of 2's compliment.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; background-color: #e6f2ff;">Programming Notes</td> </tr> <tr> <td>MBZ if scoreboard is disabled.</td> </tr> </table>	Format:	S3	Programming Notes	MBZ if scoreboard is disabled.		
	Format:	S3					
	Programming Notes						
	MBZ if scoreboard is disabled.						
<p>27:24 Scoreboard 3 Delta X</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>S3</td> </tr> </table> <p>Relative horizontal distance of the dependent instance assigned to scoreboard 3, in the form of 2's compliment.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; background-color: #e6f2ff;">Programming Notes</td> </tr> <tr> <td>MBZ if scoreboard is disabled.</td> </tr> </table>	Format:	S3	Programming Notes	MBZ if scoreboard is disabled.			
Format:	S3						
Programming Notes							
MBZ if scoreboard is disabled.							
<p>23:20 Scoreboard 2 Delta Y</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>S3</td> </tr> </table> <p>Relative vertical distance of the dependent instance assigned to scoreboard 2, in the form of 2's compliment.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; background-color: #e6f2ff;">Programming Notes</td> </tr> <tr> <td>MBZ if scoreboard is disabled.</td> </tr> </table>	Format:	S3	Programming Notes	MBZ if scoreboard is disabled.			
Format:	S3						
Programming Notes							
MBZ if scoreboard is disabled.							
<p>19:16 Scoreboard 2 Delta X</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>S3</td> </tr> </table> <p>Relative horizontal distance of the dependent instance assigned to scoreboard 2, in the form of 2's compliment.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; background-color: #e6f2ff;">Programming Notes</td> </tr> <tr> <td>MBZ if scoreboard is disabled.</td> </tr> </table>	Format:	S3	Programming Notes	MBZ if scoreboard is disabled.			
Format:	S3						
Programming Notes							
MBZ if scoreboard is disabled.							



MEDIA_VFE_STATE		
8	15:12	Scoreboard 1 Delta Y Format: S3 Relative vertical distance of the dependent instance assigned to scoreboard 1, in the form of 2's compliment. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.
	11:8	Scoreboard 1 Delta X Format: S3 Relative horizontal distance of the dependent instance assigned to scoreboard 1, in the form of 2's compliment. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.
	7:4	Scoreboard 0 Delta Y Format: S3 Relative vertical distance of the dependent instance assigned to scoreboard 0, in the form of 2's compliment. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.
	3:0	Scoreboard 0 Delta X Format: S3 Relative horizontal distance of the dependent instance assigned to scoreboard 0, in the form of 2's compliment. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.
	31:28	Scoreboard 7 Delta Y Format: S3 Relative vertical distance of the dependent instance assigned to scoreboard 7, in the form of 2's compliment. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.
	27:24	Scoreboard 7 Delta X Format: S3 Relative horizontal distance of the dependent instance assigned to scoreboard 7, in the form of 2's compliment. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.



MEDIA_VFE_STATE					
23:20	Scoreboard 6 Delta Y Format: S3 Relative vertical distance of the dependent instance assigned to scoreboard 6, in the form of 2's compliment. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.				
	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. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.			
		15:12	Scoreboard 5 Delta Y Format: S3 Relative vertical distance of the dependent instance assigned to scoreboard 5, in the form of 2's compliment. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.		
			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. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.	
				7:4	Scoreboard 4 Delta Y Format: S3 Relative vertical distance of the dependent instance assigned to scoreboard 4, in the form of 2's compliment. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> MBZ if scoreboard is disabled.
					3:0



Media Block Read MSD

MSD1R_MB - Media Block Read MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Other					
Group:	Media Block R/W					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHR</td> </tr> </table> <p>Indicates that the message requires a header.</p>	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18:14	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>04h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Media Block Read message</p>	Default Value:	04h	Format:	Opcode
	Default Value:	04h				
	Format:	Opcode				
13:11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
10:8	<p>Vertical Line Stride Override</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_VLSO</td> </tr> </table> <p>If enabled, specifies the Vertical Line Stride and Vertical Line Stride Offset override fields.</p>	Format:	MDC_VLSO			
Format:	MDC_VLSO					
7:0	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS			
Format:	MDC_BTS					



Media Block Write MSD

MSD1W_MB - Media Block Write MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Other	
Group:	Media Block R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHR Indicates that the message requires a header.
	18:14	Message Type
		Default Value: 0Ah
		Format: Opcode Media Block Write message
13:11	Reserved	
	Format: MBZ Ignored	
10:8	Vertical Line Stride Override	
	Format: MDC_VLSO If enabled, specifies the Vertical Line Stride and Vertical Line Stride Offset override fields.	
7:0	Binding Table Index	
	Format: MDC_BTS Specifies the Binding Table Index for the message	



Media Transpose Read MSD

MSD1R_TT - Media Transpose Read MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Other					
Group:	Transpose Read					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_MHR</td> </tr> </table> Indicates that the message requires a header.	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18:14	Message Type <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Default Value:</td> <td style="text-align: center;">00h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Transpose Read message	Default Value:	00h	Format:	Opcode
	Default Value:	00h				
Format:	Opcode					
13:8	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
7:0	Binding Table Index <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDC_BTS</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS			
Format:	MDC_BTS					



Memory Fence MSD

MSD_MEMFENCE - Memory Fence MSD			
Source:	DataPort 0		
Length Bias:	1		
Family:	Other		
Group:	Memory Fence		
DWord	Bit	Description	
0	19	Header Present	
		Format: MDC_MHP Indicates that the message requires a header.	
	18	Legacy Message	
		Default Value: 0h	
		Format: Opcode Legacy Message	
	17:14	Message Type	
		Default Value: 07h	
		Format: Opcode Memory Fence message	
	13	Commit	
		Format: Enable	
Specifies whether control is returned to the thread only after the fence has been honored.			
Value		Name	
Description			
1	Enabled [Default]	The commit writeback register is always required to guarantee ordering.	
0	Reserved	The commit writeback register is always required to guarantee ordering.	
12:9	L3 Flush		
	The L3 Flush control is one of the following GSYNC signals.		
	Value	Name	
	Description		
	0h	Disabled [Default]	The L3 caches are not flushed.
	08h	RW Data	Causes the L3 to flush any RW data.
	04h	Constant Data	Causes the L3 to invalidate any Constant data.
02h	Texture Data	Causes the L3 to invalidate any Texture data.	
01h	Instructions	Causes the L3 to invalidate all GPU instruction caches.	



MSD_MEMFENCE - Memory Fence MSD							
	<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">If multiple caches need to be flushed, the commands need to be sent separately.</td> </tr> <tr> <td colspan="2">When the memory fence completes, the GSYNC has been started, but may not yet be completed. To know when the GSYNC is completed, Issue any read to the L3 cache after an L3 Flush operation and wait for that data to be returned.</td> </tr> </tbody> </table>	Programming Notes		If multiple caches need to be flushed, the commands need to be sent separately.		When the memory fence completes, the GSYNC has been started, but may not yet be completed. To know when the GSYNC is completed, Issue any read to the L3 cache after an L3 Flush operation and wait for that data to be returned.	
Programming Notes							
If multiple caches need to be flushed, the commands need to be sent separately.							
When the memory fence completes, the GSYNC has been started, but may not yet be completed. To know when the GSYNC is completed, Issue any read to the L3 cache after an L3 Flush operation and wait for that data to be returned.							
8	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ				
Format:	MBZ						
7:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ				
Format:	MBZ						



MFC_AVC_PAK_OBJECT

MFC_AVC_PAK_OBJECT				
Source:	VideoCS			
Length Bias:	2			
<p>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.</p>				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	3h PARALLEL_VIDEO_PIPE	
		Format:	OpCode	
	28:27	Pipeline		
		Default Value:	2h MFC_AVC_PAK_OBJECT	
		Format:	OpCode	
	26:24	Media Command Opcode		
		Default Value:	1h AVC_ENC	
		Format:	OpCode	
	23:21	SubOpcode A		
		Default Value:	2h	
		Format:	OpCode	
	20:16	SubOpcode B		
		Default Value:	9h	
Format:		OpCode		
15:12	Reserved			
	Format:	MBZ		
11:0	DWord Length	Format:	=n Length -2	
	Value	Name		
	0009h	DWORD_COUNT_n [Default]		
	000Ah	DWORD_COUNT_n [Default]		



MFC_AVC_PAK_OBJECT						
1	31:10	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
9:0	<p>Indirect PAK-MV Data Length</p> <p>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 form). A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect PAK-MV Data Start Address field is ignored. This field must have the same alignment as the Indirect PAK-MV Data Start Address. This field must be DW aligned (since each MV is 4 bytes in size). Driver has to derived this field from MVsize (MVquantity in DXVA, exact size) *4 bytes per MV.</p>					
2	31:29	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
28:0	<p>Indirect PAK-MV Data Start Address Offset</p> <p>This field specifies the memory starting address (offset) of the MV data to be fetched into PAK Subsystem for processing. This pointer is relative to the MFC Indirect PAK-MV Object Base Address. Hardware ignores this field if indirect data is not present, i.e. the Indirect PAK-MV Data Length is set to 0. It is a Dword aligned address in all AVC encoding configuration, since each MV is 4 bytes in size.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 60%; text-align: center;">Value</th> <th style="width: 40%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,512MB)</td> <td></td> </tr> </tbody> </table>		Value	Name	[0,512MB)	
Value	Name					
[0,512MB)						
3..10	31:0	<p>Inline Data</p> <p>All the required MB level controls and parameters for encoding are captured as inline data of the MFC_AVC_PAK_OBJECT command. It has a fixed size of 8 DWs. Its definition is described in the next section.</p>				
12..23	31:0	<p>Reserved</p>				



MFC_JPEG_HUFF_TABLE_STATE

MFC_JPEG_HUFF_TABLE_STATE					
Source:	VideoCS				
Length Bias:	2				
<p>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.</p> <p>Two Huffman tables for luma and chroma will be sent to H/W, and chroma table is used for both U and V.</p>					
DWord	Bit	Description			
0	31:29	Command Type			
		Default Value:	3h PARALLEL_VIDEO_PIPE		
		Format:	OpCode		
	28:27	Pipeline			
		Default Value:	2h MFC_JPEG_HUFF_TABLE_STATE		
		Format:	OpCode		
	26:24	Media Command Opcode			
		Default Value:	7h JPEG		
Format:		OpCode			
23:21	SubOpcode A				
	Default Value:	2h Common			
	Format:	OpCode			
20:16	SubOpcode B				
	Default Value:	3h MEDIA_			
	Format:	OpCode			
15:12	Reserved				
	Format:	MBZ			
11:0	DWord Length				
	Default Value:	0AEh Excludes DWord (0,1)			
	Format:	=n Total Length - 2			
1	31:1	Reserved			
		Format:	MBZ		
0	0	Huff Table ID			
		Format:	U1		
		Huffman table destination identifier will specify one of two destinations at the encoder into which the Huffman table must be stored.			
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> </tbody> </table>	Value	Name	Description
Value	Name	Description			



MFC_JPEG_HUFF_TABLE_STATE								
		<table border="1"> <tr> <td style="width: 50px;">0</td> <td></td> <td>Huffman table 0</td> </tr> <tr> <td>1</td> <td></td> <td>Huffman table 1</td> </tr> </table>	0		Huffman table 0	1		Huffman table 1
0		Huffman table 0						
1		Huffman table 1						
2..13	31:0	<p>DC_TABLE</p> <table border="1"> <tr> <td style="width: 50px;">Format:</td> <td>3Bytes: Byte0 for Code length, Byte1 and Byte2 for Code word, and Byte3 for dummy</td> </tr> </table> <p>12 categories with code length and code word. Each run/size has 1-byte code length, and 2-byte code word.</p>	Format:	3Bytes: Byte0 for Code length, Byte1 and Byte2 for Code word, and Byte3 for dummy				
Format:	3Bytes: Byte0 for Code length, Byte1 and Byte2 for Code word, and Byte3 for dummy							
14..175	31:0	<p>AC_TABLE</p> <table border="1"> <tr> <td style="width: 50px;">Format:</td> <td>3Bytes: Byte0 for Code length, Byte1 and Byte2 for Code word, and Byte3 for dummy</td> </tr> </table> <p>162 run/size with code length and code word. Each run/size has 1-byte code length, and 2-byte code word.</p>	Format:	3Bytes: Byte0 for Code length, Byte1 and Byte2 for Code word, and Byte3 for dummy				
Format:	3Bytes: Byte0 for Code length, Byte1 and Byte2 for Code word, and Byte3 for dummy							



MFC_JPEG_SCAN_OBJECT

MFC_JPEG_SCAN_OBJECT		
Source:	VideoCS	
Length Bias:	2	
Encoder Only		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
		Format: OpCode
	28:27	Pipeline
		Default Value: 2h MFC_JPEG_SCAN_OBJECT
		Format: OpCode
	26:24	Media Command Opcode
		Default Value: 7h JPEG_ENC
Format: OpCode		
23:21	SubOpcode A	
	Default Value: 2h	
	Format: OpCode	
20:16	SubOpcode B	
	Default Value: 9h	
	Format: OpCode	
15:12	Reserved	
	Format: MBZ	
11:0	DWord Length	
	Default Value: 001h Excludes DWord (0,1)	
	Format: =n Total Length - 2	
1	31:26	Reserved
		Format: MBZ
	25:0	MCU Count
		Format: U26
<p>This field indicates the number of MCUs in the Scan.</p> <p>MCU Count = $M_x \times M_y$</p> <p>The number of MCUs in a row: $M_x = (X + (H_1 * 8 - 1)) / (H_1 * 8)$</p> <p>The number of MCUs in a column: $M_y = (Y + (V_1 * 8 - 1)) / (V_1 * 8)$</p> <p>X: The number of samples per line in Y-image</p> <p>Y: The number of lines in Y-image</p> <p>H1: Horizontal sampling factor of Y-image in the Frame header</p>		



MFC_JPEG_SCAN_OBJECT																							
		V1: Vertical sampling factor of Y-image in the Frame header																					
2	31:25	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ																			
	Format:	MBZ																					
	24:22	<p>Huffman AC Table</p> <p>AC Huffman table destination selector specifies one of two possible AC table destinations for each Y, U, V, or R, G, B. The AC Huffman tables must have been loaded in destination 0 and 1 by the time of issuing MFC_JPEG_HUFF_TABLE_STATE Command.</p> <p>If AC table 0 is used for Y and AC table 1 is used for U and V, it will be set to 110b. If AC table 0 is used for R, G, and B, it will be set to 000b and so on. Refer to the table below for the summary of actions.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0XXb</td> <td>Bit24 (V0)</td> <td>The third image component must use the AC table 0.</td> </tr> <tr> <td>1XXb</td> <td>Bit24 (V1)</td> <td>The third image component must use the AC table 1.</td> </tr> <tr> <td>X0Xb</td> <td>Bit23 (U0)</td> <td>The second image component must use the AC table 0.</td> </tr> <tr> <td>X1Xb</td> <td>Bit23 (U1)</td> <td>The second image component must use the AC table 1.</td> </tr> <tr> <td>XX0b</td> <td>Bit22 (Y0)</td> <td>The first image component must use the AC table 0.</td> </tr> <tr> <td>XX1b</td> <td>Bit22 (Y1)</td> <td>The first image component must use the AC table 1.</td> </tr> </tbody> </table> <p style="text-align: center;">Restriction</p> <p>Restriction : When InputSurfaceFormatYUV = RGB, because the order of input image components can be RGB, GBR, BGR, or YUV, Bit22 is used for the first image component, Bit23 is used for the second image component, and Bit24 is used for the third image component.</p>	Value	Name	Description	0XXb	Bit24 (V0)	The third image component must use the AC table 0.	1XXb	Bit24 (V1)	The third image component must use the AC table 1.	X0Xb	Bit23 (U0)	The second image component must use the AC table 0.	X1Xb	Bit23 (U1)	The second image component must use the AC table 1.	XX0b	Bit22 (Y0)	The first image component must use the AC table 0.	XX1b	Bit22 (Y1)	The first image component must use the AC table 1.
	Value	Name	Description																				
0XXb	Bit24 (V0)	The third image component must use the AC table 0.																					
1XXb	Bit24 (V1)	The third image component must use the AC table 1.																					
X0Xb	Bit23 (U0)	The second image component must use the AC table 0.																					
X1Xb	Bit23 (U1)	The second image component must use the AC table 1.																					
XX0b	Bit22 (Y0)	The first image component must use the AC table 0.																					
XX1b	Bit22 (Y1)	The first image component must use the AC table 1.																					
21	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ																				
Format:	MBZ																						
20:18	<p>Huffman DC Table</p> <p>DC Huffman table destination selector specifies one of two possible DC table destinations for each Y, U, V, or R, G, B. The DC Huffman tables shall have been loaded in destination 0 and 1 by the time of issuing MFC_JPEG_HUFF_TABLE_STATE Command.</p> <p>if DC table 0 is used for Y and DC table 1 is used for U and V, it will be set to 110b. If DC table 0 is used for R, G, and B, it will be set to 000b and so on. Refer to the table below for the summary of actions.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0XXb</td> <td>Bit20 (V0)</td> <td>The third image component must use the DC table 0.</td> </tr> </tbody> </table>	Value	Name	Description	0XXb	Bit20 (V0)	The third image component must use the DC table 0.																
Value	Name	Description																					
0XXb	Bit20 (V0)	The third image component must use the DC table 0.																					



MFC_JPEG_SCAN_OBJECT			
	1XXb	Bit20 (V1)	The third image component must use the DC table 1.
	X0Xb	Bit19 (U0)	The second image component must use the DC table 0.
	X1Xb	Bit19 (U1)	The second image component must use the DC table 1.
	XX0b	Bit18 (Y0)	The first image component must use the DC table 0.
	XX1b	Bit18 (Y1)	The first image component must use the DC table 1.
Restriction			
Restriction : When InputSurfaceFormatYUV = RGB, because the order of input image components can be RGB, GBR, BGR, YUV, Bit18 is used for the first image component, Bit19 is used for the second image component, and Bit20 is used for the third image component.			
17	Head Present Flag		
	If this flag is set to 0, then no MFC_JPEG_PAK_INSERT_OBJECT commands will be sent.		
	If this flag is set to 1, then one or more MFC_JPEG_PAK_INSERT_OBJECT commands will be sent after MFC_JPEG_SCAN_OBJECT command.		
	Value	Name	Description
	0		No insertion into the output bitstream buffer before Scan encoded bitstream
	1		Headers, tables, App data insertion into the output bitstream buffer. HW will insert the insertion data before the Scan encoded bitstream.
16	Is Last Scan		
	If this flag is set, then HW will insert EOI (0xFFD9) to the end of Scan encoded bitstream.		
	Value	Name	Description
	0		Not the last Scan.
	1		Indicates that the current Scan is the last one.
15:0	Restart Interval		
	Format:	U16	
	Specifies the number of MCUs in an ECS, except for the last ECS. Restart maker is inserted periodically and it separates the two neighboring ECSs.		
	Value	Name	
	0-FFFFh		
Programming Notes			
A value of '0' implies that the Scan Data has a single ECS.			



MFC_MPEG2_PAK_OBJECT

MFC_MPEG2_PAK_OBJECT			
Source:	VideoCS		
Length Bias:	2		
<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFC_AVC_PAK_INSERT_OBJECT
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	3h MPEG2
		Format:	OpCode
	23:21	SubOpcode A	
		Default Value:	2h ENC
		Format:	OpCode
	20:16	SubOpcode B	
		Default Value:	9h MEDIA_
Format:		OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	0007h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	



MFC_MPEG2_PAK_OBJECT

1..8	31:0	Inline Data All the required MB level controls and parameters for encoding are captured as inline data of the MFC_MPEG2_PAK_OBJECT command. It has a fixed size of 8 DWs. Its definition is described in the next section
------	------	---



MFC_MPEG2_SLICEGROUP_STATE

MFC_MPEG2_SLICEGROUP_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>This is a slice group level command and can be issued multiple times within a picture that is comprised of multiple slice groups. The same command is used for AVC encoder (PAK mode) and decoder (VLD and IT modes).</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_MPEG2_SLICEGROUP_STATE
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	3h MPEG2
		Format:	OpCode
	23:21	SubOpcode A	
Default Value:		2h MEDIA_	
Format:		OpCode	
20:16	SubOpcode B		
	Default Value:	3h MEDIA_	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	6h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31	MbRateCtrlFlag- RateControlCounterEnable (Encoder-only)	
		To enable the accumulation of bit allocation for rate controlThis field enables hardware Rate Control logic. The rest of the RC control fields are only valid when this field is set to 1. Otherwise, hardware ignores these fields.Note: To reset MB level rate control (QRC), we need to set both bits MbRateCtrlFlag and MbRateCtrlReset to 1 in the new slice	
		Value	Name
		0h	Disable
1h	Enable		



MFC_MPEG2_SLICEGROUP_STATE		
30	MbRateCtrlReset- ResetRateControlCounter (Encoder-only) To reset the bit allocation accumulation counter to 0 to restart the rate control.	
	Value	Name
	0h	Disable
	1h	Enable
29:28	MbRateCtrlMode- RC Triggle Mode (Encoder-only)	
	Value	Name
	00b	Always Rate Control, whereas RC becomes active if sum_act > sum_target or sum_act < sum_target
	01b	Gentle Rate Control, whereas RC becomes active if sum_act > upper_midpt or sum_act < lower_midpt
	10b	Loose Rate Control, whereas RC becomes active if sum_act > sum_max or sum_act < sum_min
	11b	Reserved
27:24	MbRateCtrlParam- RC Stable Tolerance (Encoder-only)	
	Format:	U4
	This field specifies the tolerance required to deactivate RC once it has been triggered.	
	Value	Name
	[0, 15]	
23	RateCtrlPanicFlag - RC Panic Enable (Encoder-only) If this field is set to 1, RC enters panic mode when sum_act > sum_max. RC Panic Type field controls what type of panic behavior is invoked.	
	Value	Name
	0	Disable
	1	Enable
22	RateCtrlPanicType - RC Panic Type (Encoder-only) This field selects between two RC Panic methods. 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.	
	Value	Name
	0h	QP Panic
	1h	CBP Panic
21	Reserved	
	Format:	MBZ



MFC_MPEG2_SLICEGROUP_STATE			
20	SkipConvDisabled - MB Type Skip Conversion Disable (Encoder-only)		
	This field is only valid for a P or B slice. It must be zero for other slice types. Rules are provided in Section 2.3.3.1.6		
	Value	Name	Description
	0h	Enable	Enable skip type conversion
	1h	Disable	Disable skip type conversion
19	IsLastSliceGrp		
	IsLastSliceGrp = 1 if the current slice group is the last slice group of a picture; 0 otherwise. It is used by the zero filling in the Minimum Frame Size test.		
18	BitstreamOutputFlag - Compressed BitStream Output Disable Flag (Encoder-only)		
	Value	Name	Description
	0h	Enable	enable the writing of the output compressed bitstream
	1h	Disable	disable the writing of the output compressed bitstream
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		no tail insertion into the output bitstream buffer, after the current slice encoded bits
	1h		tail insertion into the output bitstream buffer is present, and is after the current slice 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	



MFC_MPEG2_SLICEGROUP_STATE						
	7:4	SliceID[3:0] (Encoder-only) To identify the output data (coding information record) returned for rate control from PAK to ENC and VPP				
	3:2	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ for StreamID extension</td></tr></table>		MBZ for StreamID extension		
		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: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> also CurrStartVertPos, Vertical count of the first MB in the current slice group (Encoder-only)		U8		
		U8				
7:0	FirstMbXcnt - also CurrStartHorzPos Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> Horizontal count of the first MB in the current slice group (Encoder-only)		U8			
	U8					
3	31:9	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ		
		MBZ				
	8	SliceGroupSkip Exists If: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>//Encoder Only</td></tr></table> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U1</td></tr></table> All macroblocks are skipped		//Encoder Only		U1
		//Encoder Only				
	U1					
7:6	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ			
	MBZ					
5:0	SliceGroupQp Exists If: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>//Encoder Only</td></tr></table> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U6</td></tr></table> Initial slice quality parameter		//Encoder Only		U6	
	//Encoder Only					
	U6					
4	31:29	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ		
	MBZ					



MFC_MPEG2_SLICEGROUP_STATE								
	28:0	<p>BitstreamOffset - Indirect PAK-BSE Data Start Address (Write)</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> </table> <p>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.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,512MB)</td> <td></td> </tr> </tbody> </table>	Exists If:	//Encoder Only	Value	Name	[0,512MB)	
	Exists If:	//Encoder Only						
Value	Name							
[0,512MB)								
5	31:24	<p>MaxQpNegModifier - Magnitude of QP Max Negative Modifier (Encoder-only)</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>This field specifies the lower limit of the QP modifier.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, 51]</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	[0, 51]	
		Format:	U8					
		Value	Name					
	[0, 51]							
	23:16	<p>MaxQpPosModifier - Magnitude of QP Max Positive Modifier (Encoder-only)</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>This field specifies the upper limit of the QP modifier.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, 51]</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	[0, 51]	
		Format:	U8					
		Value	Name					
	[0, 51]							
	15:12	<p>ShrinkParam - Shrink Resistance (Encoder-only)</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the additional points added each time decreased correction is invoked.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, 15]</td> <td></td> </tr> </tbody> </table>	Format:	U4	Value	Name	[0, 15]	
		Format:	U4					
		Value	Name					
	[0, 15]							
11:8	<p>Shrinkaram - Shrink Init (Encoder-only)</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the initial points required to trip decreased control.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, 15]</td> <td></td> </tr> </tbody> </table>	Format:	U4	Value	Name	[0, 15]		
	Format:	U4						
	Value	Name						
[0, 15]								
7:4	<p>GrowParam - Grow Resistance (Encoder-only)</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the additional points added each time increased correction is invoked.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, 15]</td> <td></td> </tr> </tbody> </table>	Format:	U4	Value	Name	[0, 15]		
	Format:	U4						
	Value	Name						
[0, 15]								



MFC_MPEG2_SLICEGROUP_STATE						
6	3:0	GrowParam - Grow Init (Encoder-only) Format: U4 This field specifies the initial points required to trip increased control. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0, 15]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0, 15]	
	Value	Name				
	[0, 15]					
	31:24	Reserved Format: MBZ				
	23:20	CorrectPoints - Correct 6 (Encoder-only) Format: U4 This field specifies the points used in the lowermost RC region when $sum_act \leq sum_min$. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0, 15]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0, 15]	
	Value	Name				
[0, 15]						
19:16	CorrectPoints - Correct 5 (Encoder-only) Format: U4 This field specifies the points used in the fifth RC region when $sum_act > sum_min$ but $\leq lower_midpt$. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0, 15]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0, 15]		
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 $\leq sum_target$. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0, 15]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0, 15]		
Value	Name					
[0, 15]						
11:8	CorrectPoints - Correct 3 (Encoder-only) Format: U4 This field specifies the points used in the third RC region when $sum_act > sum_target$ but $\leq upper_midpt$. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0, 15]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0, 15]		
Value	Name					
[0, 15]						
7:4	CorrectPoints - Correct 2 (Encoder-only) Format: U4 This field specifies the points used in the second RC region when $sum_act > upper_midpt$ but $\leq sum_max$. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0, 15]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0, 15]		
Value	Name					
[0, 15]						



MFC_MPEG2_SLICEGROUP_STATE																																										
	3:0	CorrectPoints - Correct 1 (Encoder-only)																																								
	<table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U4</td> </tr> <tr> <td colspan="2">This field specifies the points used in the topmost RC region when sum_act > sum_max</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>[0, 15]</td> <td></td> </tr> </table>		Format:	U4	This field specifies the points used in the topmost RC region when sum_act > sum_max		Value	Name	[0, 15]																																	
Format:	U4																																									
This field specifies the points used in the topmost RC region when 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 Only																																									
	Format: U4																																									
	23:20	CV5 - Clamp Value 5 (Encoder-only)																																								
	Exists If: //Encoder Only																																									
	Format: U4																																									
19:16	CV4 - Clamp Value 4 (Encoder-only)																																									
Exists If: //Encoder Only																																										
Format: U4																																										
15:12	CV3 - Clamp Value 3 (Encoder-only)																																									
Exists If: //Encoder Only																																										
Format: U4																																										
11:8	CV2 - Clamp Value 2 (Encoder-only)																																									
Exists If: //Encoder Only																																										
Format: U4																																										
7:4	CV1 - Clamp Value 1 (Encoder-only)																																									
Exists If: //Encoder Only																																										
Format: U4																																										
3:0	CV0 - Clamp Value 0 (Encoder-only) If the magnitude of coefficients at locations assigned with CV0 (mapping shown below) exceeds 2CV0-1, they are replaced with 2CV0-1. For coefficients at locations marked as 'none', no clamping is performed. The following mappings are only applied to luma and chroma blocks\subblocks containing AC coefficients (blocks\subblocks with only DC coeffs will not be clamped). For 8x8 frame block, each coefficient is mapped to one of the eight CV values as following: <table border="1" style="margin-top: 10px;"> <tr><td>none</td><td>none</td><td>CV7</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td></tr> <tr><td>none</td><td>CV7</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td></tr> <tr><td>CV7</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td><td>CV2</td></tr> <tr><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td><td>CV2</td><td>CV1</td></tr> <tr><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td><td>CV2</td><td>CV1</td><td>CV1</td></tr> </table>		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
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																																			



MFC_MPEG2_SLICEGROUP_STATE

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 8x8 field block, each coefficient is mapped to one of the eight CV values as following:

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

MFD_AVC_BSD_OBJECT			
Source:	VideoCS		
Length Bias:	2		
Description			
<p>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.</p>			
Context switch interrupt is not supported by this command.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFD_AVC_BSD_OBJECT
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	1h AVC_DEC
		Format:	OpCode
	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		
	Format:	=n Total Length - 2	
	Value	Name	
4h	Excludes DWord (0,1) = 0004 [Default]		



MFD_AVC_BSD_OBJECT										
1	31:0	Indirect BSD Data Length Format: U32 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. AVC Short Format : 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 after 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.								
		Reserved Format: MBZ								
2	31:29	Reserved Format: MBZ								
		28:0 Indirect BSD Data Start Address Format: U29 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. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,512MB)</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,512MB)					
Value	Name									
[0,512MB)										
3.5	31:0	Inline Data All the required Slice Header parameters and error handling settings are captured as InLine Data of the AVC_BSD_OBJECT command. It has a fixed size of 4 DWs. Its definition is described in the following section: Inline Data Description.								
6	31:13	Reserved Format: MBZ								
		12:9 Reserved								
	8	AVC Nal Type First Byte Override Bit Format: U1 This bit indicates hardware should use the NAL Type (provided below) programmed by driver instead of using the one from bitstream. The NAL byte from bitstream will not be correct. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Use Bitstream Decoded NAL Type</td> <td>NAL Type should come from the first byte of Decoded Bitstream</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Use Driver Programmed NAL Type</td> <td>NAL Type should come from "Driver Provided Nal Type" programmed by driver.</td> </tr> </tbody> </table>	Value	Name	Description	0	Use Bitstream Decoded NAL Type	NAL Type should come from the first byte of Decoded Bitstream	1	Use Driver Programmed NAL Type
Value	Name	Description								
0	Use Bitstream Decoded NAL Type	NAL Type should come from the first byte of Decoded Bitstream								
1	Use Driver Programmed NAL Type	NAL Type should come from "Driver Provided Nal Type" programmed by driver.								



MFD_AVC_BSD_OBJECT				
7:0	<p>Driver Provided Nal Type Value</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>This will replace the first byte of the NAL unit, containing forbidden_zero_bit, nal_ref_idc, and nal_unit_type.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This byte should be ignored if "AVC NAL Type First Byte Override Bit" is programmed to 0</p>	Format:	U8	Programming Notes
Format:	U8			
Programming Notes				



MFD_AVC_DPB_STATE

MFD_AVC_DPB_STATE							
Source:	VideoCS						
Length Bias:	2						
<p>This is a frame level state command used only in DXVA2 AVC Short Slice Bitstream Format VLD mode. RefFrameList[16] of DXVA2 interface is replaced with intel Reference Picture Addresses[16] of MFX_PIPE_BUF_ADDR_STATE command. The LongTerm Picture flag indicator of all reference pictures are collected into LongTermPic_Flag[16].</p> <p>FieldOrderCntList[16][2] and CurrFieldOrderCnt[2] of DXVA2 interface are replaced with intel POCList[34] of MFX_AVC_DIRECTMODE_STATE command.</p>							
DWord	Bit	Description					
0	31:29	Command Type					
		Default Value:	3h PARALLEL_VIDEO_PIPE				
		Format:	OpCode				
	28:27	28:27	Pipeline				
			Default Value:	2h MFX_MULTI_DW			
			Format:	OpCode			
	26:24	26:24	Media Command Opcode				
			Default Value:	1h AVC_DEC			
			Format:	OpCode			
	23:21	23:21	SubOpcode A				
			Default Value:	1h			
			Format:	OpCode			
	20:16	20:16	SubOpcode B				
			Default Value:	6h			
			Format:	OpCode			
	15:12	15:12	Reserved				
			Format:	MBZ			
	11:0	11:0	DWord Length				
			Format:	=n Total Length - 2			
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">9h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>		Value	Name	9h	Excludes DWord (0,1) [Default]
		Value	Name				
9h	Excludes DWord (0,1) [Default]						



MFD_AVC_DPB_STATE																	
1	31:16	<p>LongTermFrame_Flag[16][1 bit] One-to-one correspondence with the entries of the Intel RefFrameList[16]. 1 bit per reference frame.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td></td> <td>the picture is a long term reference picture</td> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>the picture is a short term reference picture</td> </tr> </tbody> </table>	Value	Name	Description	1		the picture is a long term reference picture	0		the picture is a short term reference picture						
	Value	Name	Description														
1		the picture is a long term reference picture															
0		the picture is a short term reference picture															
15:0	<p>Non-ExistingFrame_Flag[16][1 bit] One-to-one correspondence with the entries of the Intel RefFrameList[16]. 1 bit per reference frame.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>INVALID</td> <td>the reference picture in that entry of RefFrameList[] does not exist anymore.</td> </tr> <tr> <td style="text-align: center;">0</td> <td>VALID</td> <td>the reference picture in that entry of RefFrameList[] is a valid reference</td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p style="text-align: center;">Programming Notes</p> <p>When an element of the list of frames is not relevant (e.g., due to the corresponding reference entry being empty or being marked as "not used for reference"), the value of the corresponding bit of NonExistingFrameFlags shall be set to 0.</p> </div>		Value	Name	Description	1	INVALID	the reference picture in that entry of RefFrameList[] does not exist anymore.	0	VALID	the reference picture in that entry of RefFrameList[] is a valid reference						
Value	Name	Description															
1	INVALID	the reference picture in that entry of RefFrameList[] does not exist anymore.															
0	VALID	the reference picture in that entry of RefFrameList[] is a valid reference															
2	31:0	<p>UsedForReference_Flag[16][2 bits] One-to-one correspondence with the entries of the Intel RefFrameList[16]. 2 bits per reference frame.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>NOT_REFERENCE</td> <td>indicates a frame is "not used for reference".</td> </tr> <tr> <td style="text-align: center;">1</td> <td>TOP_FIELD</td> <td>bit[0] indicates that the top field of a frame is marked as "used for reference".</td> </tr> <tr> <td style="text-align: center;">2</td> <td>BOTTOM_FIELD</td> <td>bit[1] indicates that the bottom field of a frame is marked as "used for reference".</td> </tr> <tr> <td style="text-align: center;">3</td> <td>FRAME</td> <td>bit[1:0] indicates that a frame (or field pair) is marked as "used for reference".</td> </tr> </tbody> </table>	Value	Name	Description	0	NOT_REFERENCE	indicates a frame is "not used for reference".	1	TOP_FIELD	bit[0] indicates that the top field of a frame is marked as "used for reference".	2	BOTTOM_FIELD	bit[1] indicates that the bottom field of a frame is marked as "used for reference".	3	FRAME	bit[1:0] indicates that a frame (or field pair) is marked as "used for reference".
		Value	Name	Description													
		0	NOT_REFERENCE	indicates a frame is "not used for reference".													
		1	TOP_FIELD	bit[0] indicates that the top field of a frame is marked as "used for reference".													
		2	BOTTOM_FIELD	bit[1] indicates that the bottom field of a frame is marked as "used for reference".													
3	FRAME	bit[1:0] indicates that a frame (or field pair) is marked as "used for reference".															
<p>Programming Notes</p> <p>When an element of the list of frames is not relevant (e.g., due to the corresponding reference entry being empty or being marked as "not used for reference"), the value of the corresponding bit of UsedForReferenceFlags shall be set to 0.</p>																	
<p>Programming Notes</p> <p>When an element of the list of frames is not relevant (e.g., due to the corresponding reference entry being empty or being marked as "not used for reference"), the value of the corresponding bit of UsedForReferenceFlags shall be set to 0.</p>																	
<p>Programming Notes</p> <p>When an element of the list of frames is not relevant (e.g., due to the corresponding reference entry being empty or being marked as "not used for reference"), the value of the corresponding bit of UsedForReferenceFlags shall be set to 0.</p>																	
<p>Programming Notes</p> <p>When an element of the list of frames is not relevant (e.g., due to the corresponding reference entry being empty or being marked as "not used for reference"), the value of the corresponding bit of UsedForReferenceFlags shall be set to 0.</p>																	
3..10	31:0	<p>LTSTFrameNumList[16][16 bits] One-to-one correspondence with the entries of the Intel RefFrameList[16]. 16 bits per reference frame. Depending on the corresponding LongTermFrame_Flag[], the content of this field is interpreted differently.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>LongTermFrame_Flag[i]</td> <td>LTSTFrameNumList[i] represent LongTermFrameIdx.</td> </tr> <tr> <td style="text-align: center;">0</td> <td>ShortTermFrame_Flag[i]</td> <td>LTSTFrameNumList[i] represent Short Term Picture FrameNum.</td> </tr> </tbody> </table>	Value	Name	Description	1	LongTermFrame_Flag[i]	LTSTFrameNumList[i] represent LongTermFrameIdx.	0	ShortTermFrame_Flag[i]	LTSTFrameNumList[i] represent Short Term Picture FrameNum.						
		Value	Name	Description													
		1	LongTermFrame_Flag[i]	LTSTFrameNumList[i] represent LongTermFrameIdx.													
0	ShortTermFrame_Flag[i]	LTSTFrameNumList[i] represent Short Term Picture FrameNum.															
<p>Programming Notes</p> <p>When an element of the list of frames is not relevant (e.g., due to the corresponding reference entry being empty or being marked as "not used for reference"), the value of the LTSTFrameNumList entry shall be set to 0.</p>																	
<p>Programming Notes</p> <p>When an element of the list of frames is not relevant (e.g., due to the corresponding reference entry being empty or being marked as "not used for reference"), the value of the LTSTFrameNumList entry shall be set to 0.</p>																	



MFD_AVC_DPB_STATE		
11..18	31:0	<p>ViewIDList[16][16 bits] One-to-one correspondence with the entries of the Intel RefFrameList[16]. 16 bits per reference frame. The view ids are 10-bits, the upper 6 bits are ignored. "000000" & ViewId1[9:0] & "000000" & ViewId0[9:0]</p> <p style="text-align: center;">Programming Notes</p> <p>When an Intel RefFrameList[i] is not an valid entries, Viewid should be set to 0x00</p>
19..22	31:0	<p>ViewOrderListL0[16][8 bits] One-to-one correspondence with the entries of the Intel RefFrameList[16]. 8 bits per reference frame. The view order need 4-bits, the upper 4 bits are ignored. 0000 & ViewOrder3[3:0] & 0000 & ViewOrder2[3:0] & 0000 & ViewOrder1[3:0] & 0000 & ViewOrder0[3:0]</p> <p style="text-align: center;">Programming Notes</p> <p>When the ViewOrderListL0[i] is not an valid inter-view reference, its corresponding ViewOrder should be set to 0xF</p> <p>Since only interview with the same polarity will be used, there is no need to have field bit in this list. Hardware is going to append correct polarity bit as needed.</p>
23..26	31:0	<p>ViewOrderListL1[16][8 bits] One-to-one correspondence with the entries of the Intel RefFrameList[16]. 8 bits per reference frame. The view order need 4-bits, the upper 4 bits are ignored. 0000 & ViewOrder3[3:0] & 0000 & ViewOrder2[3:0] & 0000 & ViewOrder1[3:0] & 0000 & ViewOrder0[3:0]</p> <p style="text-align: center;">Programming Notes</p> <p>When the ViewOrderListL1[i] is not an valid inter-view reference, its corresponding ViewOrder should be set to 0xF</p> <p>Since only interview with the same polarity will be used, there is no need to have field bit in this list. Hardware is going to append correct polarity bit as needed.</p>



MFD_AVC_PICID_STATE

MFD_AVC_PICID_STATE		
Source:	VideoCS	
Length Bias:	2	
<p>This is a frame level state command used for both AVC Long and Short Format in VLD mode. PictureID[16] contains the pictureID of each reference picture (16 maximum) so hardware can uniquely identify the reference picture across frames (this will be used for DMV operation). This command will be needed for both short and long format.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE Format: OpCode
	28:27	Pipeline
		Default Value: 2h MFX_MULTI_DW Format: OpCode
	26:24	Media Command Opcode
		Default Value: 1h MFD_AVC_DPB_STATE Format: OpCode
	23:21	SubOpcode A
		Default Value: 1h DEC Format: OpCode
	20:16	SubOpcode B
		Default Value: 5h MEDIA_ Format: OpCode
15:12	Reserved	
11:0	DWord Length	
	Default Value: 0008h Excludes DWord (0,1) Format: =n Total Length - 2	
	Reserved	
1	31:1	Reserved
		Format: MBZ



MFD_AVC_PICID_STATE											
	0	PictureID Remapping Disable									
		<table border="1"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>0h</td><td>AVC decoder will use 16 bits Picture ID to handle DMV and identify the reference picture</td><td>Desc</td></tr><tr><td>1h</td><td>AVC decoder will use 4 bits FrameStoreID (index to RefFrameList) to handle DMV and identify the reference picture.</td><td>Desc</td></tr></tbody></table>	Value	Name	Description	0h	AVC decoder will use 16 bits Picture ID to handle DMV and identify the reference picture	Desc	1h	AVC decoder will use 4 bits FrameStoreID (index to RefFrameList) to handle DMV and identify the reference picture.	Desc
		Value	Name	Description							
		0h	AVC decoder will use 16 bits Picture ID to handle DMV and identify the reference picture	Desc							
		1h	AVC decoder will use 4 bits FrameStoreID (index to RefFrameList) to handle DMV and identify the reference picture.	Desc							
Programming Notes											
If Picture ID Remapping Disable is "1", PictureIDList will not be used.											
2..9	31:0	PictureIDList[16][16 bits] One-to-one correspondence with the entries of the Intel RefFrameList[16]. 16 bits per reference frame. PictureID of each Frame uniquely identifies the reference picture across frames. The same number cannot be reused until the reference picture is completely retired (no longer used for reference). When an element of the list of frames is not relevant (e.g., due to the corresponding reference entry being empty or being marked as "not used for reference"), the value of the LTSTFrameNumList entry shall be set to 0.									



MFD_AVC_SLICEADDR

MFD_AVC_SLICEADDR			
Source:	VideoCS		
Length Bias:	2		
<p>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 concealment 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 be 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 $256 \times 256 = 64K-1$ 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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFD_AVC_SLICEADDR
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	1h AVC_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	



MFD_AVC_SLICEADDR										
	11:0	DWord Length Format: =n Total Length - 2 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>	Value	Name	1h	Excludes DWord (0,1) [Default]				
		Value	Name							
		1h	Excludes DWord (0,1) [Default]							
1	31:0	Indirect BSD Data Length Format: U32 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 after 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. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 60%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,512MB)</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,512MB)					
Value	Name									
[0,512MB)										
3	31:13	Reserved Format: MBZ								
	12:9	Reserved								
	8	AVC NAL Type First Byte Override Bit Format: U1 This bit indicates hardware should use the NAL Type (provided below) programmed by driver instead of using the one from bitstream. The NAL byte from bitstream will not be correct. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 30%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Use Bitstream Decoded NAL Type</td> <td>NAL Type should come from first byte of decoded bitstream.</td> </tr> <tr> <td>1</td> <td>Use Driver Programmed NAL Type</td> <td>NAL Type should come from "Driver Provided NAL Type Values" programmed by driver.</td> </tr> </tbody> </table>	Value	Name	Description	0	Use Bitstream Decoded NAL Type	NAL Type should come from first byte of decoded bitstream.	1	Use Driver Programmed NAL Type
Value	Name	Description								
0	Use Bitstream Decoded NAL Type	NAL Type should come from first byte of decoded bitstream.								
1	Use Driver Programmed NAL Type	NAL Type should come from "Driver Provided NAL Type Values" programmed by driver.								



MFD_AVC_SLICEADDR				
7:0	<p>Driver Provided NAL Type Value</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>This will replace the first byte of the NAL unit, containing forbidden_zero_bit, nal_ref_idc, and nal_unit_type.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This byte should be ignored if "AVC NAL Type First Byte Override Bit" is programmed to 0</p>	Format:	U8	Programming Notes
Format:	U8			
Programming Notes				



MFD_IT_OBJECT

MFD_IT_OBJECT			
Source:	VideoCS		
Length Bias:	2		
All weight mode (default and implicit) are mapped to explicit mode. But the weights come in either as explicit or implicit.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFD_IT_OBJECT
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	0h MFX_COMMON_DEC
		Format:	OpCode
	23:21	SubOpcode A	
Default Value:		1h	
Format:		OpCode	
20:16	SubOpcode B		
	Default Value:	9h	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	06h Excludes DWord (0,1) For AVC = Ch	
	Format:	=n Total Length - 2 Note: Regardless of the mode, inline data must be present in this command.	
1	31:10	Reserved	
		Format:	MBZ



MFD_IT_OBJECT					
	9:0	<p>Indirect IT-MV Data Length</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U10 FormatDesc: In bytes</td> </tr> </table> <p>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 form). A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect IT-MV Data Start Address field is ignored. This field must have the same alignment as the Indirect Object Data Start Address.AVC-IT Mode: It must be DWord aligned (since each MV is 4bytes in size)Driver has to derived this field from MVsize (MVquantity in DXVA, exact size) *4 bytes per MV.This field is only valid in AVC decoder IT mode (VC1 and MPEG uses inline MV data).</p>	Format:	U10 FormatDesc: In bytes	
Format:	U10 FormatDesc: In bytes				
2	31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
	Format:	MBZ			
28:0	<p>Indirect IT-MV Data Start Address Offset</p> <p>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).</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,512MB)</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,512MB)	
Value	Name				
[0,512MB)					
3	31:12	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
	Format:	MBZ			
11:0	<p>Indirect IT-COEFF Data Length</p> <p>This field provides the length in bytes of the indirect data, which contains all the non-zero coefficients for the current MB. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect IT-COEFF Data Start Address field is ignored. Since each IT-COEFF data is 1 DW in size, with 12 bits, this field can be extended to support up to 4:4:4 format.(256 pixel * 3 byte pixel components * 4 bytes per coeff).This field must be integer multiple of 16-bytes for AVC (since each coefficient is 4 bytes in size).This field is only valid in AVC, VC1, MPEG2 decoder IT mode.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 40%;">Value</th> <th style="width: 60%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,3072]</td> <td>In bytes [0, 256*3*4]</td> </tr> </tbody> </table>	Value	Name	[0,3072]	In bytes [0, 256*3*4]
Value	Name				
[0,3072]	In bytes [0, 256*3*4]				
4	31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ				



MFD_IT_OBJECT						
	<p>28:0 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 coefficient 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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,512MB)</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,512MB)		
Value	Name					
[0,512MB)						
5	<p>31:6 Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ				
<p>5:0 Indirect IT-DBLK Control Data Length</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U6</td> </tr> </table> <p>This field provides the length in bytes of the indirect data, which contains all the deblocker control information for the current MB (in 4x4 sub-block partitioning). A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect IT-DBLK Data Start Address field is ignored. This field must have the same alignment as the Indirect IT-DBLK Data Start Address. 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.</p>	Format:	U6				
Format:	U6					
6	<p>31:29 Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ				
<p>28:0 Indirect IT-DBLK Control Data Start Address Offset</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">Format:</td> <td>IndirectObjectBaseAddress[28:0]</td> </tr> </table> <p>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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,512MB)</td> <td></td> </tr> </tbody> </table>	Format:	IndirectObjectBaseAddress[28:0]	Value	Name	[0,512MB)	
Format:	IndirectObjectBaseAddress[28:0]					
Value	Name					
[0,512MB)						
7..n	<p>31:0 Inline Data Union for all 3 codecs Includes IT, MC, IntraPred inline data as well as Deblocker control information AVC-IT Modes: Hardware interprets this data in the specified format. VC1-IT Modes: Hardware interprets this data in the specified format. MV inline MPEG2-IT Modes: Hardware interprets this data in the specified format. (IS mode) MV inline For AVC there 7 DWords of inline data, hence N is equal to 13.</p>					



MFD_JPEG_BSD_OBJECT

MFD_JPEG_BSD_OBJECT			
Source:	VideoCS		
Length Bias:	2		
Exists If:	//Decoder		
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFD_JPEG_BSD_OBJECT
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	7h JPEG_DEC
Format:		OpCode	
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 DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:0	Indirect Data Length . It is the length in bytes of the bitstream data for the current Scan. It includes the first byte of the first MCU and the last non-zero byte of the last MCU in the Scan. Specifically, the zero-padding bytes (if present) are excluded. Hardware ignores the contents after the last non-zero byte.	
2	31:29	Reserved	
		Format:	MBZ



MFD_JPEG_BSD_OBJECT											
	28:0	<p>Indirect Data Start Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>IndirectObjectOffset[28:0]</td> </tr> </table> <p>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 BSD Indirect Object Base Address. Hardware ignores this field if indirect data is not present. It is a byte-aligned address for the JPEG bitstream data</p>	Format:	IndirectObjectOffset[28:0]							
Format:	IndirectObjectOffset[28:0]										
3	31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	28:16	<p>Scan Horizontal Position</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U13 bits in blocks</td> </tr> </table> <p>This field indicates the horizontal position (in block units) of the first MCU in the Scan.</p>	Format:	U13 bits in blocks							
	Format:	U13 bits in blocks									
15:13	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
12:0	<p>Scan Vertical Position</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U13 bits in blocks</td> </tr> </table> <p>This field indicates the vertical position (in block units) of the first MCU in the Scan.</p>	Format:	U13 bits in blocks								
Format:	U13 bits in blocks										
4	31	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	30	<p>Interleaved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Non-Interleaved</td> <td>one component in the Scan</td> </tr> <tr> <td>1</td> <td>Interleaved</td> <td>multiple components in the Scan</td> </tr> </tbody> </table>	Value	Name	Description	0	Non-Interleaved	one component in the Scan	1	Interleaved	multiple components in the Scan
	Value	Name	Description								
	0	Non-Interleaved	one component in the Scan								
1	Interleaved	multiple components in the Scan									
29:27	<p>Scan Components</p> <p>Bit0: Y Bit1: U Bit2: V</p> <p>For example, if non-interleaved Y, then it will be set to 001b. If interleaved Y, U, and V, it will be set to 111b.</p>										
26	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
25:0	<p>MCU Count</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U26</td> </tr> </table> <p>This field indicates the number of MCUs in the Scan.</p>	Format:	U26								
Format:	U26										
5	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ										



MFD_JPEG_BSD_OBJECT			
15:0	<p>RestartInterval(16 bit)</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U16</td> </tr> </table> <p>Specifies the number of MCU in restart interval. Valid values are 1->0xFFFFValue of 0 implies that all the SCAN have only one ECS.</p>	Format:	U16
Format:	U16		



MFD_MPEG2_BSD_OBJECT

MFD_MPEG2_BSD_OBJECT			
Source:	VideoCS		
Length Bias:	2		
<p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFD_MPEG2_BSD_OBJECT
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	3h MPEG2_DEC
		Format:	OpCode
	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 DWord (0,1)	
	Format:	=n Total Length - 2	



MFD_MPEG2_BSD_OBJECT							
1	31:0	<p>Indirect BSD Data Length</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U32</td> </tr> </table> <p>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.</p> <p>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. So the maximum slice size for 4K x 4K is $4608 * 256 / 8 = 147,456$ bytes (0x24000), which requires 18 bits.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>As MPEG-2 spec does not post any limitation of the size of zero-padding bytes, it is 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</td> </tr> <tr> <td>zero-padding restriction is removed</td> </tr> </table>	Format:	U32	Programming Notes	As MPEG-2 spec does not post any limitation of the size of zero-padding bytes, it is 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	zero-padding restriction is removed
Format:	U32						
Programming Notes							
As MPEG-2 spec does not post any limitation of the size of zero-padding bytes, it is 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							
zero-padding restriction is removed							
2	31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ						
	28:0	<p>Indirect Data Start Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">IndirectObjectOffset[28:0]</td> </tr> </table> <p>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 BSD Indirect Object Base Address. Hardware ignores this field if indirect data is not present. It is a byte-aligned address for the MPEG2 VLD bitstream data This address points to the first byte of the MB layer data, i.e. not including slice header.</p>	Format:	IndirectObjectOffset[28:0]			
Format:	IndirectObjectOffset[28:0]						
3.4	31:0	<p>Inline Data</p> <p>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 definition is described in the next section.</p>					



MFD_VC1_BSD_OBJECT

MFD_VC1_BSD_OBJECT		
Source:	VideoCS	
Length Bias:	2	
<p>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.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
	Format: OpCode	
	28:27	Pipeline
		Default Value: 2h MFX_MULTI_DW
	Format: OpCode	
	26:24	Media Command Opcode
		Default Value: 2h VC1_DEC
Format: OpCode		
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 DWord (0,1)	
Format: =n Total Length - 2		
1	31:24	Reserved
		Format: MBZ



MFD_VC1_BSD_OBJECT									
	23:0	<p>Indirect BSD Data Length</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U24</td> </tr> </table> <p>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. GEN6 Long Format : It is the length in bytes of the bitstream data for the current slice/picture. It includes the first byte of the first macroblock and the last byte of the last macroblock in the slice/picture. Specifically, the zero-padding bytes (if present) and the next start-code are excluded. Hardware ignores the contents after the last non-zero byte (trailing zeros). This field is sized to support VC1 AP@L4 Level bitstream. It includes the byte that contains the First MB Bit Offset GEN7 Short Format : It is the length in bytes of the bitstream data for the current slice, including Picture/Slice Header + Emulation 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.</p>	Format:	U24					
	Format:	U24							
2	<p>31:29 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>28:0 Indirect Data Start Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>IndirectObjectOffset[28:0]</td> </tr> </table> <p>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 VC1 bitstream data.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 60%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,512MB)</td> <td></td> </tr> </tbody> </table>	Format:	MBZ	Format:	IndirectObjectOffset[28:0]	Value	Name	[0,512MB)	
Format:	MBZ								
Format:	IndirectObjectOffset[28:0]								
Value	Name								
[0,512MB)									
3	31:24	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
	23:16	<p>Slice Start Vertical Position</p> <p>This field specifies the position in y-direction of the first macroblock in the Slice in unit of macroblocks. For SecondField this value is reset to zero as oppoed to the VC1 spec Ref: 9.1.2 Slice Layer. This field is for both Long and Short VC1 Interface Format.</p>							
	15:9	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ								
8:0	<p>Next Slice Vertical Position</p> <p>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 the last 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.</p>								
4	31:16	<p>First_MB_Byte_Offset_of_Slice_Data_or_Slice_Header</p> <p>For DXVA2 VC1 Short Format only It gives the byte offset to locate the first MB data in the bitstream for a slice, relative to the Indirect BSD Data Start Address.</p>							



MFD_VC1_BSD_OBJECT				
	15:5	Reserved		
		Format: MBZ		
	4	Emulation Prevention Byte Present		
		Value	Name Description	
		0h		H/W needs to perform Emulation Byte Removal
		1h		H/W does not need to perform Emulation Byte Removal
	3	Reserved		
		Format: MBZ		
	2:0	FirstMbBitOffset (First Macroblock Bit Offset)		
		Format: U3		
This field provides the bit offset of the first macroblock of the Slice in the first byte of the input compressed bitstream.It is used with First_MB_Byte_Offset for non-byte aligned position.				



MFD_VC1_LONG_PIC_STATE

MFD_VC1_LONG_PIC_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>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. This Long interface format is intel proprietary interface. Driver will need to perform addition operations to generate all the fields in this command.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFD_VC1_LONG_PIC_STATE
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	2h VC1_DEC
		Format:	OpCode
	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 DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:24	Reserved	
		Format:	MBZ



MFD_VC1_LONG_PIC_STATE											
	<p>23:16 PictureHeightInMBsMinus1 (Picture Height Minus 1 in Macroblocks)</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>This field indicates the height of the picture in unit of macroblocks. 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.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2"> <p>Note: Even though the Advanced Profile allows frame dimensions (width, height) to not be aligned to macroblock boundary, it doesn't affect the bitstream decoding. And it is preferable to use 'intermediate buffer' that is macroblock aligned for decoding. In order to simplify the out-of-bound reference pixel access, the out-of-bound extrapolation rule in VC1 spec can be used to expand the expected decoded frame to the intermediate buffer dimension.</p> </td> </tr> </table>	Format:	U8	Programming Notes		<p>Note: Even though the Advanced Profile allows frame dimensions (width, height) to not be aligned to macroblock boundary, it doesn't affect the bitstream decoding. And it is preferable to use 'intermediate buffer' that is macroblock aligned for decoding. In order to simplify the out-of-bound reference pixel access, the out-of-bound extrapolation rule in VC1 spec can be used to expand the expected decoded frame to the intermediate buffer dimension.</p>					
	Format:	U8									
	Programming Notes										
<p>Note: Even though the Advanced Profile allows frame dimensions (width, height) to not be aligned to macroblock boundary, it doesn't affect the bitstream decoding. And it is preferable to use 'intermediate buffer' that is macroblock aligned for decoding. In order to simplify the out-of-bound reference pixel access, the out-of-bound extrapolation rule in VC1 spec can be used to expand the expected decoded frame to the intermediate buffer dimension.</p>											
<p>15:8 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ										
<p>7:0 PictureWidthInMBsMinus1 (Picture Width Minus 1 in Macroblocks)</p> <table border="1"> <tr> <td>Format:</td> <td>U8-1</td> </tr> </table> <p>This field indicates the width of the picture in unit of macroblocks. For example, for a 1920x1080 frame picture, PictureWidthInMBs equals 120 (1920 divided by 16). This field is used in VLD and IT modes</p>	Format:	U8-1									
Format:	U8-1										
2	<p>31:24 Bitplane Buffer Pitch Minus 1</p> <table border="1"> <tr> <td>Format:</td> <td>U7-1 Pitch in (Bytes - 1).</td> </tr> </table> <p>Specifies the bitplane buffer pitch in (#Bytes - 1). Bitplane buffer is a linear buffer. It is needed only when the bitplane is not encoded as raw, and therefore is present in the header explicitly. In VC1 Long Format (Gen6 and Gen7), it is written by an application and later read by the HW. But in VC1 Short Format (Gen7 only), it is written and read by H/W only. This field is specified for better performance</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, FFFFFFFFh]</td> <td></td> </tr> </tbody> </table> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2"> <p>For Gen6 : The pitch must be equal to PictureWidthInMBs/2. For Gen7 VC1 Long Format : The pitch must be equal to PictureWidthInMBs/2. For Gen7 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 mode, used in VLD mode only. For VC1 DXVA2 Short Format, the bitplane specification is between H/W and Driver only. For Long Format, application is responsible for allocation with the driver.</p> </td> </tr> </table>	Format:	U7-1 Pitch in (Bytes - 1).	Value	Name	[0, FFFFFFFFh]		Programming Notes		<p>For Gen6 : The pitch must be equal to PictureWidthInMBs/2. For Gen7 VC1 Long Format : The pitch must be equal to PictureWidthInMBs/2. For Gen7 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 mode, used in VLD mode only. For VC1 DXVA2 Short Format, the bitplane specification is between H/W and Driver only. For Long Format, application is responsible for allocation with the driver.</p>	
	Format:	U7-1 Pitch in (Bytes - 1).									
	Value	Name									
[0, FFFFFFFFh]											
Programming Notes											
<p>For Gen6 : The pitch must be equal to PictureWidthInMBs/2. For Gen7 VC1 Long Format : The pitch must be equal to PictureWidthInMBs/2. For Gen7 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 mode, used in VLD mode only. For VC1 DXVA2 Short Format, the bitplane specification is between H/W and Driver only. For Long Format, application is responsible for allocation with the driver.</p>											
<p>23:16 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ										



MFD_VC1_LONG_PIC_STATE

15	DmvSurfaceValid	<p>Indicated when the DMV read surface is valid. This surface stored the direct motion vectors and Mb type. This field is set for B pictures that can refer to a previous P picture for DMV. If there is an I-picture before a B (in decoding order) then this field is not set (as a result, zero's DMV's will be assumed while decoding the B picture. That is, there is no explicit DMV buffer for an I-picture). When the current picture being decoded is an I, P or BI, this bit is set to 0, since there is no DMV read in these picture decoding process. This field is not used in IT mode, used in VLD mode only.</p>															
14	ImplicitQuantizer	<p>Derived by driver from QUANTIZER. This field is used in intel 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</p>															
13	Interpolation Runder Contro	<p>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.</p>															
Programming Notes																	
This bit field is taken from bRcontrol in DXVA_PictureParameters data structure																	
12	SyncMarker	<p>Indicates whether sync markers are enabled/disabled. If enable, sync markers "may be" present in the current video sequence being decoded. It is a sequence level syntax element and is valid only for Simple and Main Profiles.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Not Present</td> <td>Sync Marker is not present in the bitstream</td> </tr> <tr> <td>1h</td> <td>Maybe present</td> <td>Sync Marker maybe present in the bitstream</td> </tr> </tbody> </table>	Value	Name	Description	0h	Not Present	Sync Marker is not present in the bitstream	1h	Maybe present	Sync Marker maybe present in the bitstream						
Value	Name	Description															
0h	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 Main Profile, SyncMarker can be set to 0 or 1. This field is used in both intel and MS DXVA2 VLD interface, but not used in IT mode.																	
11:8	Motion Vector Mode	<p>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. 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.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0XX0b</td> <td></td> <td>Chroma Quarter -pel + Luma bicubic. (can only be 1MV)</td> </tr> <tr> <td>0XX1b</td> <td></td> <td>Chroma Half-pel + Luma bicubic. (can be 1MV or 4MV)</td> </tr> <tr> <td>1XX0b</td> <td></td> <td>Chroma Quarter -pel + Luma bilinear. (can only be 1MV)</td> </tr> <tr> <td>1XX1b</td> <td></td> <td>Chroma Half-pel + Luma bilinear</td> </tr> </tbody> </table>	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
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															

**MFD_VC1_LONG_PIC_STATE****Programming Notes**

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 MC. Bit 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 down on-the-fly, if RangeReduction is Enabled.

Value	Name	Description
0h		Scale down reference picture by factor of 2
1h		Scale up reference picture by factor of 2

Programming Notes

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 only. Driver is responsible to keep RangeReductionScale, RangeReduction Enable and RANGERED Present Flag of current picture coherent.

6

RangeReduction Enable

This 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.

Value	Name	Description
0h	Disable	Range reduction is not performed
1h	Enable	Range reduction is performed

Programming Notes

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 only. Driver is responsible to keep RangeReductionScale, RangeReduction Enable and RANGERED Present Flag of current picture coherent.



MFD_VC1_LONG_PIC_STATE		
5	LOOPFILTER Enable Flag This field is the decoded syntax element LOOPFILTER in bitstream. It indicates if In-loop Deblocking is ON according to picture level bitstream syntax control. This bit affects BSD unit and also the loop filter unit. When this bit is set to 1, PostDeblockOutEnable field in MFX_PIPE_MODE_SELECT command must also be set to 1. In this case, in-loop deblocking operation follows the VC1 standard - deblocking doesn't cross slice boundary. When this bit is set to 0, but PostDeblockOutEnable field in MFX_PIPE_MODE_SELECT command is set to 1. It indicates the loop filter unit is used for out-of-loop deblocking. In this case, deblocking operation does cross slice boundary. This field is used in VLD mode only, not in IT mode.	
	Value	Name
	0h	Disable
	1h	Enable
4	Overlap Smoothing Enable Flag This field is the decoded syntax element OVERLAP in bitstream. Indicates if Overlap smoothing is ON at the picture level. This field is used in both VLD and IT modes.	
	Value	Name
	0h	Disable
	1h	Enable
3	Secondfield This flag is set for the second field in field pictures. This field is used in both VLD and IT modes.	
2:1	Reserved	
	Format:	MBZ
0	VC1 Profile specifies the bitstream profile. This field is used in both VLD and IT modes.	
	Value	Name
	0h	Disable
	1h	Enable
	Programming Notes	
This is required because 128 is added for intra blocks post inverse transform in advanced profile and also to find out if Motion vectors are adjusted or not.		
3	31	Reserved
		Format:
	30:29	CondOver This field is the decoded syntax element CONDOVER in a bitstream of advanced profile. It controls the overlap smoothing filter operation for an I frame or an BI frame when the picture level quantization step size PQUANT is 8 or lower. This field is used in intel VC1 VLD mode only, not in DXVA2 VC1 and IT modes.
Value		Name
Description		



MFD_VC1_LONG_PIC_STATE		
	00b	No overlap smoothing
	01b	Reserved
	10b	Always perform overlap smoothing filter
	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 Mode. When FCM = 00 01 (a Progressive or Interlaced Frame Picture): 000 = I001 = P010 = B011 = BI100 = Skipped. Other encodings are reserved. When FCM = 10 11 (a Field Picture): 000 = I/I001 = I/P010 = P/I011 = P/P100 = B/B101 = B/B110 = BI/B111 = BI/BIA. Although, for a field picture, it is set for a field-pair, but HW will only look at one field state only, and the other field state is don't care. This field is read and qualified with the SecondField flag internally. This field is unique to intel VC1 VLD Long format, and is used in IT mode as well. For DXVA2 VC1 IT mode, driver needs to convert the DXVA2 interface to intel HW VLD Long Format interface.	
25:24	FCM (Frame Coding Mode) This is the same as the variable FCM defined in VC1. This field must be set to 0 for Simple and Main Profiles. This field is unique to intel VC1 VLD Long format, and is used in IT mode as well. For DXVA2 VC1 IT mode, driver needs to convert the DXVA2 interface to intel HW VLD Long Format interface.	
	Value	Name
	00b	Disable
	01b	Enable
	10b	Field Picture with Top Field First
	11b	Field Picture with Bottom Field First
23:21	Reserved Format: MBZ	
20:16	AltPQuant (Alternative Picture Quantization Value) This field is identical to the variable ALTPQUANT which is derived from VOPDQUANT configuration in the VC1 standard. This field must be set to 0 for Simple/Main I and BI pictures as VOPDQUANT is not present. This field is used in intel VC1 VLD Long Format mode only, not used in DXVA2 VC1 VLD and IT modes.	
15:13	Reserved 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 as PQuant = (PQINDEX < 29) ? PQINDEX - 3 : PQINDEX*2 - 31. This field is used in all picture types (I, P, B and BI) and all operating modes (IT mode and intel and DXVA2 VLD modes).	



MFD_VC1_LONG_PIC_STATE																	
7:0	<p>BScaleFactor BScaleFactor 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. 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 intel VC1 VLD Long format mode, it is not used in DXVA2 VC1 VLD and IT modes. BFRACTION VLCBFRACTIONBScaleFactor0001/21280011/3850102/31700111/4641003/41921011/5511102/510211100003/515311100014/520411100101/64311100115/621511101001/73711101012/77411101103/711111101114/714811110005/718511110016/722211110101/83211110113/89611111005/816011111017/8224</p>																
4	31:30	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
	29:28	<p>UnifiedMvMode (Unified Motion Vector Mode) This field is a combination of the variables MVMODE and MVMODE2 in the VC1 standard, for parsing Luma MVD from the bitstream. This field is used to signal 1MV vs 4MV allowed (Mixed Mode). This field is also used to signal Q-pel or Half-pel MVD read from the bitstream. The bicubic or bilinear Luma MC interpolation mode is duplicate information from Motion Vector Mode field, and is ignored here. This field is used in intel VC1 VLD Long Format mode only, it is not used in DXVA2 VC1 VLD and IT modes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td></td> <td>Mixed MV, Q-pel bicubic</td> </tr> <tr> <td>01b</td> <td></td> <td>1-MV, Q-pel bicubic</td> </tr> <tr> <td>10b</td> <td></td> <td>1-MV half-pel bicubic</td> </tr> <tr> <td>11b</td> <td></td> <td>1-MV half-pel bilinear</td> </tr> </tbody> </table>	Value	Name	Description	00b		Mixed MV, Q-pel bicubic	01b		1-MV, Q-pel bicubic	10b		1-MV half-pel bicubic	11b		1-MV half-pel bilinear
Value	Name	Description															
00b		Mixed MV, Q-pel bicubic															
01b		1-MV, Q-pel bicubic															
10b		1-MV half-pel bicubic															
11b		1-MV half-pel bilinear															
	27	<p>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 intel VC1 VLD Long Format mode only, it is not used in DXVA2 VC1 VLD and IT modes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>only 1-MV</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>1, 2, or 4 MVs</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	only 1-MV	1h	Enable	1, 2, or 4 MVs						
Value	Name	Description															
0h	Disable	only 1-MV															
1h	Enable	1, 2, or 4 MVs															
	26	<p>FastUVMCFlag (Fast UV Motion Compensation Flag) This field specifies whether the motion vectors for UV is rounded to half or full pel position. It is identical to the variable FASTUVMC in VC1 standard. This field is used in both VLD and IT modes. It is derived from FASTUVMC = (bPicSpatialResid8 » 4) & 1 in both VLD and IT modes, and should have the same value as Motion Vector Mode LSBit.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>no rounding</td> </tr> </tbody> </table>	Value	Name	Description	0h		no rounding									
Value	Name	Description															
0h		no rounding															



MFD_VC1_LONG_PIC_STATE											
	1h	quarter-pel offsets to half/full pel positions									
25	<p>RefFieldPicPolarity (Reference Field Picture Polarity) This field specifies the polarity of the one reference field picture used for a field P picture. It is derived from the variable REFFIELD defined in VC1 standard and is only valid when one field is referenced (NUMREF = 0) for a field P picture. When NUMREF = 0 and REFFIELD = 0, this field is the polarity of the reference I/P field that is temporally closest; When NUMREF = 0 and REFFIELD = 1, this field is the polarity of the reference I/P field that is the second most temporally closest. The distance is measured based on display order but ignoring the repeated field if present (due to RFF = 1). This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Top (even) field</td> </tr> <tr> <td>1h</td> <td></td> <td>Bottom (odd) field</td> </tr> </tbody> </table>		Value	Name	Description	0h		Top (even) field	1h		Bottom (odd) field
Value	Name	Description									
0h		Top (even) field									
1h		Bottom (odd) field									
24	<p>NumRef (Number of References) This field indicates how many reference fields are referenced by the current (field) picture. It is identical to the variable NUMREF in the VC1 standard. This field is only valid for field P picture (FCM = 10 11). This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>One field referenced</td> </tr> <tr> <td>1h</td> <td></td> <td>Two fields referenced</td> </tr> </tbody> </table>		Value	Name	Description	0h		One field referenced	1h		Two fields referenced
Value	Name	Description									
0h		One field referenced									
1h		Two fields referenced									
23:20	<p>BwdRefDist (Reference Distance) This field is valid only in B field pictures giving the value of BRFD. The field is ignored in P Picture. This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p>										
19:16	<p>FwdRefDist (Reference Distance) Format: <table border="1" style="display: inline-table;"><tr><td>U4</td></tr></table> This field is the number of frames between the current frame and its reference frame. It is derived 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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0, 15]</td> <td></td> </tr> </tbody> </table>		U4	Value	Name	[0, 15]					
U4											
Value	Name										
[0, 15]											
15:12	<p>Reserved Format: <table border="1" style="display: inline-table;"><tr><td>MBZ</td></tr></table></p>		MBZ								
MBZ											
11:10	<p>ExtendedDMVRange (Extended Differential Motion Vector Range Flag) This field specifies the differential motion vector range in interlaced pictures. It is equivalent to the variable DMVRANGE in the VC1 standard. This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p>										



MFD_VC1_LONG_PIC_STATE			
	Value	Name	Description
	00b		No extended range
	01b		Extended horizontally
	10b		Extended vertically
	11b		Extended in both directions
9:8	ExtendedMVRRange (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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes		
	Value	Name	Description
	00b		[-256, 255] x [-128, 127]
	01b		512, 511] x [-256, 255]
	10b		[-2048, 2047] x [-1024, 1023]
	11b		[-4096, 4095] x [-2048, 2047]
7:4	AltPQuantEdgeMask (Alternative Picture Quantization Edge Mask) This field is a bit mask for the four edges in clock-wise order, indicating whether AltPQuant is used for the edge macroblocks. 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 found.. This field is valid only if AltPQuantConfig is 01. Bit 0: Left picture edge macroblocks Bit 1: Top picture edge macroblocks Bit 2: Right picture edge macroblocks Bit 3: Bottom picture edge macroblocks This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.		
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, MQANT. 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 found.. This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.		
	Value	Name	Description
	00b		AltPQuant not used
	01b		AltPQuant is used and applied to edge macroblocks only
	10b		MQANT is encoded in macroblock layer
	11b		AltPQuant and PQuant are selected on macroblock basis
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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.		
0	PQuantUniform Indicating if uniform quantization applies to the picture. It is used for inverse quantization of the AC coefficients. QUANTIZER 001123PQUANTIZER - -01--PQINDEX>=9<=8----		



MFD_VC1_LONG_PIC_STATE											
		<p>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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Non-uniform</td> </tr> <tr> <td>1h</td> <td></td> <td>Uniform</td> </tr> </tbody> </table>	Value	Name	Description	0h		Non-uniform	1h		Uniform
Value	Name	Description									
0h		Non-uniform									
1h		Uniform									
5	31	<p>BitplanePresentFlag (Bitplane Buffer Present Flag) This field indicates whether the bitplane buffer is present for the picture. If set, at least one of the fields listed in bits 22:16 is coded in non-raw mode, and Bitplane Buffer Base Address field in the VC1_BSD_BUF_BASE_STATE command points to the bitplane buffer. Otherwise, all the fields that are applicable for the current picture in bits 22:16 must be coded in raw mode.This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>bitplane buffer is not present</td> </tr> <tr> <td>1h</td> <td></td> <td>bitplane buffer is present</td> </tr> </tbody> </table>	Value	Name	Description	0h		bitplane buffer is not present	1h		bitplane buffer is present
	Value	Name	Description								
	0h		bitplane buffer is not present								
	1h		bitplane buffer is present								
30	<p>ForwardMbRaw This field indicates whether the FORWARDMB field is coded in raw or non-raw mode.This field is only valid when PictureType is B.This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>non-raw mode</td> </tr> <tr> <td>1h</td> <td></td> <td>raw mode</td> </tr> </tbody> </table>	Value	Name	Description	0h		non-raw mode	1h		raw mode	
Value	Name	Description									
0h		non-raw mode									
1h		raw mode									
29	<p>MvTypeMbRaw This field indicates whether the MVTYPREMB field is coded in raw or non-raw mode.This field is only valid when PictureType is P.This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Non-Raw Mode</td> </tr> <tr> <td>1h</td> <td></td> <td>Raw Mode</td> </tr> </tbody> </table>	Value	Name	Description	0h		Non-Raw Mode	1h		Raw Mode	
Value	Name	Description									
0h		Non-Raw Mode									
1h		Raw Mode									
28	<p>SkipMbRaw This field indicates whether the SKIPMB field is coded in raw or non-raw mode.This field is only valid when PictureType is P or B.0 = non-raw mode1 = raw modeThis field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Non-Raw Mode</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Raw Mode</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	Non-Raw Mode	1h	Enable	Raw Mode	
Value	Name	Description									
0h	Disable	Non-Raw Mode									
1h	Enable	Raw Mode									



MFD_VC1_LONG_PIC_STATE										
27	<p>DirectMbRaw</p> <p>This field indicates whether the DIRECTMB field is coded in raw or non-raw mode. This field is only valid when PictureType is P or B. This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Non-Raw Mode</td> </tr> <tr> <td>1h</td> <td></td> <td>Raw Mode</td> </tr> </tbody> </table>	Value	Name	Description	0h		Non-Raw Mode	1h		Raw Mode
Value	Name	Description								
0h		Non-Raw Mode								
1h		Raw Mode								
26	<p>OverflagsRaw</p> <p>This field indicates whether the OVERFLAGS 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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Non-Raw Mode</td> </tr> <tr> <td>1h</td> <td></td> <td>Raw Mode</td> </tr> </tbody> </table>	Value	Name	Description	0h		Non-Raw Mode	1h		Raw Mode
Value	Name	Description								
0h		Non-Raw Mode								
1h		Raw Mode								
25	<p>AcPredRaw</p> <p>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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Non-Raw Mode</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Raw Mode</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	Non-Raw Mode	1h	Enable	Raw Mode
Value	Name	Description								
0h	Disable	Non-Raw Mode								
1h	Enable	Raw Mode								
24	<p>FieldTxRaw</p> <p>This field indicates whether the FIELDTX 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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Non-Raw Mode</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Raw Mode</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	Non-Raw Mode	1h	Enable	Raw Mode
Value	Name	Description								
0h	Disable	Non-Raw Mode								
1h	Enable	Raw Mode								
23	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
22:20	<p>MvTab (Motion Vector Table)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U3</td> </tr> </table> <p>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 pictures 0 = Motion Vector Differential VLD Table 01 = Motion Vector Differential VLD Table 12 = Motion Vector Differential VLD Table 23 = Motion Vector Differential VLD Table 3 The other encodings are reserved For P interlace field pictures with NUMREF = 0 or P/B interlace frame</p>	Format:	U3							
Format:	U3									



MFD_VC1_LONG_PIC_STATE

	<p>pictures0 = 1-Reference Table 01 = 1-Reference Table 12 = 1-Reference Table 23 = 1-Reference Table 34 The other encodings are reservedFor P interlace field picture with NUMREF = 1 or B interlaced field pictures0 = 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 78 The other encodings are reservedThis field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p>										
19:18	<p>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 = 4MVBP Table 12 = 4MVBP Table 23 = 4MVBP Table 34 This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p>										
17:16	<p>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 34 This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p>										
15:14	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>		Format:	MBZ							
Format:	MBZ										
13:12	<p>TransType (Picture-level Transform Type)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">U2</td> </tr> </table> <p>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 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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p>		Format:	U2							
Format:	U2										
11	<p>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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #D9E1F2;"> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td></td> <td>variable transform type in macroblock layer</td> </tr> <tr> <td style="text-align: center;">1h</td> <td></td> <td>use picture level transform type TransType</td> </tr> </tbody> </table>		Value	Name	Description	0h		variable transform type in macroblock layer	1h		use picture level transform type TransType
Value	Name	Description									
0h		variable transform type in macroblock layer									
1h		use picture level transform type TransType									



MFD_VC1_LONG_PIC_STATE											
10:8	<p>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 30 other encodings are invalid Three bits are defined for interlace field P, B pictures. There are two set of code tables selected based on if 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 7 This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p>										
7:6	<p>TransAcY (Picture-level Transform Luma AC Coding Set Index, TRANSACTABLE2) BitFieldDesc</p>										
5:4	<p>TransAcUV (Picture-level Transform Chroma AC Coding Set Index, TRANSACTABLE) This field, together with PQINDEX, specifies which intra AC coding set to be used for decoding the non-zero AC coefficients in a coded luma (Y) block. This field is the combination of the variables TRANSACFRM and TRANSACFRM2 in the VC1 standard. For I pictures, TransAcY is the same as TRANSACFRM2. For other pictures, it is the same as TRANSACFRM, and therefore must be programmed to be the same as TransAcUV. This field is valid for all picture types. 0 = Coding set index 01 = Coding set index 12 = Coding set index 23 is invalid This field is unique to intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p>										
3	<p>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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td></td> <td>The high motion tables</td> </tr> <tr> <td style="text-align: center;">1h</td> <td></td> <td>The low motion tables</td> </tr> </tbody> </table>		Value	Name	Description	0h		The high motion tables	1h		The low motion tables
Value	Name	Description									
0h		The high motion tables									
1h		The low motion tables									
2:0	<p>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 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 intel VC1 VLD Long format mode, and is not used in IT and DXVA2 VC1 modes.</p>										



MFD_VC1_SHORT_PIC_STATE

MFD_VC1_SHORT_PIC_STATE		
Source:		VideoCS
Length Bias:		2
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
		Format: OpCode
	28:27	Pipeline
		Default Value: 2h MFD_VC1_SHORT_PIC_STATE
		Format: OpCode
	26:24	Media Command Opcode
		Default Value: 2h VC1_DEC
Format: OpCode		
23:21	SubOpcode A	
	Default Value: 1h	
	Format: OpCode	
20:16	SubOpcode B	
	Default Value: 0h	
	Format: OpCode	
15:12	Reserved	
Format: MBZ		
11:0	DWord Length	
	Default Value: 0003h Excludes DWord (0,1)	
	Format: =n Total Length - 2	
1	31:24	Reserved
	Format: MBZ	
1	23:16	Picture Height
	Format: U8-1 Picture Height in Macroblocks	
<p>This field indicates the height of the picture in unit of macroblocks. 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 boundary, it doesn't affect the bitstream decoding. And it is preferable to use 'intermediate buffer' that is macroblock aligned for decoding. In order to simplify the out-of-bound reference pixel access, the out-of-bound extrapolation rule in VC1 spec can be used to expand the expected decoded frame to the intermediate buffer dimension.</p>		



MFD_VC1_SHORT_PIC_STATE			
	<p>15:8 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ	
<p>7:0 Picture Width</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U8-1 Picture Width in Macroblocks</td> </tr> </table> <p>This field indicates the width of the picture in unit of macroblocks. For example, for a 1920x1080 frame picture, PictureWidthInMBs equals 120 (1920 divided by 16). This field is used in VLD and IT modes.</p>	Format:	U8-1 Picture Width in Macroblocks	
Format:	U8-1 Picture Width in Macroblocks		
2	<p>31:24 Bitplane Buffer Pitch Minus 1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>U7-1 Pitch in Bytes</td> </tr> </table> <p>Specifies the bitplane buffer pitch in (#Bytes - 1). Bitplane buffer is a linear buffer. It is needed only when the bitplane is not encoded as raw, and therefore is present in the header explicitly. In VC1 Long Format (Gen6 and Gen7), it is written by an application and later read by the HW. In VC1 Long Format (Gen6 and Gen7), it is written by an application, and later read by the HW. But in VC1 Short Format (Gen7 only), it is written and read by H/W only. This field is specified for better performance. For Gen6 : The pitch must be equal to PictureWidthInMBs/2. For Gen7 VC1 Long Format : The pitch must be equal to PictureWidthInMBs/2. For Gen7 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 mode, used in VLD mode only. For VC1 DXVA2 Short Format, the bitplane specification is between H/W and Driver only. For Long Format, application is responsible for allocation with the driver.</p>	Format:	U7-1 Pitch in Bytes
	Format:	U7-1 Pitch in Bytes	
	<p>23 Interpolation Rounder Control</p> <p>Used only in MC operation. This field specifies the rounding control value used in interpolation operation of motion prediction process. Note: This bit field is taken from bRcontrol in DXVA_PictureParameters data structure This field is used in VLD and IT modes.</p>		
	<p>22:20 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
<p>19:16 Motion Vector Mode</p> <p>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. (can only be 1MV) 0XX1 = Chroma Half-pel + Luma bicubic. (can be 1MV or 4MV) 1XX0 = Chroma Quarter -pel + Luma bilinear. (can only be 1MV) 1XX1 = Chroma Half-pel + Luma bilinear. Note: Bits 19:16 are taken from bMVprecisionAndChromaRelation in DXVA_PictureParameters data structure. Bit 19 of Motion Vector Mode = 1 for Luma Bilinear MC; = 0 for Luma Bicubic MC. Bit 16 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. 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. ???</p>			



MFD_VC1_SHORT_PIC_STATE											
15	<p>DmvSurfaceValid Indicated when the DMV read surface is valid. This surface stored the direct motion vectors. This field is set for B pictures that can refer to a previous P picture for DMV. If there is an I-picture before a B (in decoding order) then this field is not set (as a result, zero's DMV's will be assumed while decoding the B picture. That is, there is no explicit DMV buffer for an I-picture). This field is not used in IT mode, used in VLD mode only.</p>										
14:12	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>		Format:	MBZ							
Format:	MBZ										
11	<p>VC1 Profile specifies the bitstream profile. Note: This is required because 128 is added for intra blocks post inverse transform in advanced profile and also to find out if Motion vectors are adjusted or not. This field is used in both VLD and IT modes.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 75%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">[Default]</td> <td>current picture is in Simple or Main Profile (No need to distinguish Simple and Main Profile)</td> </tr> <tr> <td style="text-align: center;">1h</td> <td></td> <td>current picture is in Advanced Profile</td> </tr> </tbody> </table>		Value	Name	Description	0h	[Default]	current picture is in Simple or Main Profile (No need to distinguish Simple and Main Profile)	1h		current picture is in Advanced Profile
Value	Name	Description									
0h	[Default]	current picture is in Simple or Main Profile (No need to distinguish Simple and Main Profile)									
1h		current picture is in Advanced Profile									
10:6	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>		Format:	MBZ							
Format:	MBZ										
5	<p>Backward Prediction Present Flag Note : a B picture that only uses forward prediction may have this flag set to 1 as well. Driver may still need to provide a valid reference picture index. This field is used in both DXVA2 VC1 VLD mode and IT mode. It is the same parameter as bPicBackwardPrediction 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.</p>										
4	<p>Intra Picture Flag This field is used in both DXVA2 VC1 VLD mode and IT mode. It is the same parameter as bPicIntra 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.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 75%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td></td> <td>entire picture can have a mixture of intra and inter MB type or just inter MB type.</td> </tr> <tr> <td style="text-align: center;">1h</td> <td></td> <td>entire picture is coded in intra MB type</td> </tr> </tbody> </table>		Value	Name	Description	0h		entire picture can have a mixture of intra and inter MB type or just inter MB type.	1h		entire picture is coded in intra MB type
Value	Name	Description									
0h		entire picture can have a mixture of intra and inter MB type or just inter MB type.									
1h		entire picture is coded in intra MB type									
3	<p>SecondField This flag is set for the second field in field pictures. This field is used in both VLD and IT modes.</p>										



MFD_VC1_SHORT_PIC_STATE																	
	2	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	1:0	<p>Picture Structure</p> <p>This field is used in both DXVA2 VC1 VLD mode and IT mode. It is the same parameter as bPicStructure in DXVA2 VC1 spec.</p> <p>The Picture Structure and Progressive Pic Type are used to derive the picture structure as specified in FCM, in DXVA2 VC1 VLD and IT mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>01b</td> <td></td> <td>top field (bit 0)</td> </tr> <tr> <td>10b</td> <td></td> <td>bottom field (bit 1)</td> </tr> <tr> <td>11b</td> <td></td> <td>frame (both fields are present)</td> </tr> <tr> <td>00b</td> <td></td> <td>illegal</td> </tr> </tbody> </table>	Value	Name	Description	01b		top field (bit 0)	10b		bottom field (bit 1)	11b		frame (both fields are present)	00b		illegal
	Value	Name	Description														
	01b		top field (bit 0)														
10b		bottom field (bit 1)															
11b		frame (both fields are present)															
00b		illegal															
3	31	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
	30	<p>Overlap Smoothing Enable Flag</p> <p>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</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>to disable overlap smoothing filter</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>to enable overlap smoothing filter</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	to disable overlap smoothing filter	1h	Enable	to enable overlap smoothing filter						
Value	Name	Description															
0h	Disable	to disable overlap smoothing filter															
1h	Enable	to enable overlap smoothing filter															
	29	<p>Range Reduction Scale</p> <table border="1"> <tr> <td>Access:</td> <td>None</td> </tr> </table> <p>This field specifies whether the reference picture pixel values should be scaled up or scaled 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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable [Default]</td> <td>Scale down reference picture by factor of 2</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Scale up reference picture by factor of 2</td> </tr> </tbody> </table>	Access:	None	Value	Name	Description	0h	Disable [Default]	Scale down reference picture by factor of 2	1h	Enable	Scale up reference picture by factor of 2				
Access:	None																
Value	Name	Description															
0h	Disable [Default]	Scale down reference picture by factor of 2															
1h	Enable	Scale up reference picture by factor of 2															
	28	<p>Range Reduction Enable</p> <p>This 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</p>															



MFD_VC1_SHORT_PIC_STATE																	
		<p>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 only. Driver is responsible to keep RangeReductionScale, RangeReduction Enable and RANGERED Present Flag of current picture coherent.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable [Default]</td> <td>Range reduction is not performed</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Range reduction is performed</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable [Default]	Range reduction is not performed	1h	Enable	Range reduction is performed						
Value	Name	Description															
0h	Disable [Default]	Range reduction is not performed															
1h	Enable	Range reduction is performed															
27:24	Reserved	<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
23:22	Progressive Pic Type	<p>This field is used in both DXVA2 VC1 VLD mode and IT mode. It is the same parameter as bPicExtrapolation in DXVA2 VC1 spec. The Picture Structure and Progressive Pic Type are used to derive the picture structure as specified in FCM, in DXVA2 VC1 VLD and IT mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>progressive only picture</td> </tr> <tr> <td>1</td> <td></td> <td>progressive only picture</td> </tr> <tr> <td>2</td> <td></td> <td>interlace picture (frame-interlace or field-interlace)</td> </tr> <tr> <td>3</td> <td></td> <td>illegal</td> </tr> </tbody> </table>	Value	Name	Description	0		progressive only picture	1		progressive only picture	2		interlace picture (frame-interlace or field-interlace)	3		illegal
Value	Name	Description															
0		progressive only picture															
1		progressive only picture															
2		interlace picture (frame-interlace or field-interlace)															
3		illegal															
21	Reserved	<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
20:16	P-Pic Ref Distance	<table border="1"> <tr> <td>Access:</td> <td>None</td> </tr> </table> <p>This element defines the number of frames between the current frame and the reference frame. It is the same as the REFDIST SE in VC1 interlaced field picture header. It is present if the entry-level flag REFDIST_FLAG == 1, and if the picture type is not one of the following types: B/B, B/BI, BI/B, BI/BI. If the entry level flag REFDIST_FLAG == 0, REFDIST shall be set to the default value of 0. This field is used in DXVA2 VC1 VLD mode only, not used in IT and intel VC1 VLD Long Format modes.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0-16</td> <td>unsigned integer</td> </tr> <tr> <td>0h</td> <td>[Default]</td> </tr> </tbody> </table>	Access:	None	Value	Name	0-16	unsigned integer	0h	[Default]							
Access:	None																
Value	Name																
0-16	unsigned integer																
0h	[Default]																



MFD_VC1_SHORT_PIC_STATE			
15:14	QUANTIZER		
	Value	Name	Description
	00b		implicit quantizer at frame level
	01b		explicit quantizer at frame level, and use PQUANTIZER SE to specify uniform or non-uniform
	10b		explicit quantizer, and non-uniform quantizer for all frames
	11b		explicit quantizer, and uniform quantizer for all frames
13	MULTIRES Present Flag (for Simple/Main Profile only)		
	Value	Name	Description
	0h		RESPIC Parameter is present in the picture header
	1h		RESPIC Parameter is present in the picture header
12	SYNCMARKER Present Flag (for Simple/Main Profile only)		
	Value	Name	Description
	0		Bitstream for Simple and Main Profile has no sync marker
	1		Bitstream for Simple and Main Profile may have sync marker(s)
11	RANGERED Present Flag (for Simple/Main Profile only)		
	It is needed for Picture Header Parsing.Driver is responsible to keep RangeReductionScale, RangeReduction Enable and RANGERED Present Flag of current picture coherent.		
	Value	Name	Description
	0		Range Reduction Parameter (RANGEREDFRM) is not present in the picture header
	1		Range Reduction Parameter (RANGEREDFRM) is present in the picture header.
10:8	MAXBFRAMES Number of consecutive B Frames.		
7	PANSCAN Present Flag		
	Value	Name	Description
	0		Pan Scan Parameters are not present in the picture header
	1		Pan Scan Parameters are present in the picture header
6	REFDIST_FLAG For header parsing REFDIST.This is used in DXVA2 VC1 VLD mode only, not used in IT and intel VC1 VLD modes.		
5	LOOPFILTER Enable Flag This filed is the decoded syntax element LOOPFILTER in bitstream. It indicates if In-loop Deblocking is ON according to picture level bitstream syntax control. This bit affects BSD unit and also the loop filter unit.When this bit is set to 1, PostDeblockOutEnable field in MFX_PIPE_MODE_SELECT command must also be set to 1. In this case, in-loop deblocking		



MFD_VC1_SHORT_PIC_STATE																								
		<p>operation follows the VC1 standard - deblocking doesn't cross slice boundary. When this bit is set to 0, but PostDeblockOutEnable field in MFX_PIPE_MODE_SELECT command is set to 1. It indicates the loop filter unit is used for out-of-loop deblocking. In this case, deblocking operation does cross slice boundary. This field is used in VLD mode only, not in IT mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>In-Loop-Deblocking-Filter is disabled</td> </tr> <tr> <td>1</td> <td></td> <td>In-Loop-Deblocking-Filter is enabled</td> </tr> </tbody> </table>	Value	Name	Description	0		In-Loop-Deblocking-Filter is disabled	1		In-Loop-Deblocking-Filter is enabled													
Value	Name	Description																						
0		In-Loop-Deblocking-Filter is disabled																						
1		In-Loop-Deblocking-Filter is enabled																						
4	<p>FastUVMCFlag (Fast UV Motion Compensation Flag) This field specifies whether the motion vectors for UV is rounded to half or full pel position. It is identical to the variable FASTUVMC in VC1 standard. This field is used in both VLD and IT modes. It is derived from $FASTUVMC = (bPicSpatialResid8 \gg 4) \& 1$ in both VLD and IT modes, and should have the same value as Motion Vector Mode LSBit.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>no rounding</td> </tr> <tr> <td>1h</td> <td></td> <td>quarter-pel offsets to half/full pel positions</td> </tr> </tbody> </table>	Value	Name	Description	0h		no rounding	1h		quarter-pel offsets to half/full pel positions														
Value	Name	Description																						
0h		no rounding																						
1h		quarter-pel offsets to half/full pel positions																						
3	<p>EXTENDED_MV Present Flag BitFieldDesc</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Extended_MV is not present in the picture header</td> </tr> <tr> <td>1h</td> <td></td> <td>Extended_MV is present in the picture header</td> </tr> </tbody> </table>	Value	Name	Description	0h		Extended_MV is not present in the picture header	1h		Extended_MV is present in the picture header														
Value	Name	Description																						
0h		Extended_MV is not present in the picture header																						
1h		Extended_MV is present in the picture header																						
2:1	<p>DQUANT</p> <table border="1"> <tr> <td>Access:</td> <td>None</td> </tr> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Use for Picture Header Parsing of VOPDUANT elements</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>[Default]</td> <td></td> </tr> <tr> <td>00b</td> <td></td> <td>no VOPDQUANT elements; Quantizer cannot vary in frame, same quantization step size PQUANT is used for all MBs in the frame</td> </tr> <tr> <td>01b</td> <td></td> <td>refer to VC1 Spec. for all the MB position dependent quantizer selection</td> </tr> <tr> <td>10b</td> <td></td> <td>The macroblocks located on the picture edge boundary shall be quantized with ALTPQUANT while the rest of the macroblocks shall be quantized with PQUANT.</td> </tr> <tr> <td>11b</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Access:	None	Format:	U2	Value	Name	Description	0h	[Default]		00b		no VOPDQUANT elements; Quantizer cannot vary in frame, same quantization step size PQUANT is used for all MBs in the frame	01b		refer to VC1 Spec. for all the MB position dependent quantizer selection	10b		The macroblocks located on the picture edge boundary shall be quantized with ALTPQUANT while the rest of the macroblocks shall be quantized with PQUANT.	11b	Reserved		
Access:	None																							
Format:	U2																							
Value	Name	Description																						
0h	[Default]																							
00b		no VOPDQUANT elements; Quantizer cannot vary in frame, same quantization step size PQUANT is used for all MBs in the frame																						
01b		refer to VC1 Spec. for all the MB position dependent quantizer selection																						
10b		The macroblocks located on the picture edge boundary shall be quantized with ALTPQUANT while the rest of the macroblocks shall be quantized with PQUANT.																						
11b	Reserved																							



MFD_VC1_SHORT_PIC_STATE				
	0	VSTRANSFORM flag		
		Value	Name	Description
		0h	Disable	variable-sized transform coding is not enabled
		1h	Enable	variable-sized transform coding is enabled
4	31:29	Reserved		
		Format: MBZ (for possible 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 Intel VC1 VLD Long Format mode and IT mode. BFRACTION VLCBFRACTION		
		Enum 0001/200011/310102/320111/431003/441011/551102/5611100003/5711100014/5811100101/6911100115/61011101001/71111101012/71211101103/71311101114/71411110005/71511110016/71611110101/81711110113/81811111005/81911111017/8201111111BI Pic Indicator31 (optional)		
	23	Reserved		
		Format: MBZ Advanced Profile only; RANGE_MAPY_FLAG Range Mapping not supported		
	22:20	Reserved		
		Format: MBZ Advanced Profile only; RANGE_MAPY Range Mapping not supported		
	19	Reserved		
		Format: MBZ Advanced Profile only; RANGE_MAPUV_FLAG Range Mapping not supported		
	18:16	Reserved		
		Format: MBZ Advanced Profile only; RANGE_MAPUV Range Mapping not supported		
15:9	Reserved			
	Format: MBZ			
8	4MV Allowed Flag			
7	POSTPROC Flag			
6	PULLDOWN			
5	INTERLACE			
4	TFCNTRFLAG			
3	FINTERFLAG			

**MFD_VC1_SHORT_PIC_STATE**

	2	REFPIC Flag For a BI picture, REFPIC flag must set to 0. For 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.									
		<table border="1"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>0h</td><td></td><td>the current picture after decoded, will never used as a reference picture</td></tr><tr><td>1h</td><td></td><td>the current picture after decoded, will be used as a reference picture later</td></tr></tbody></table>	Value	Name	Description	0h		the current picture after decoded, will never used as a reference picture	1h		the current picture after decoded, will be used as a reference picture later
	Value	Name	Description								
	0h		the current 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										
	<table border="1"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>0h</td><td>[Default]</td><td>Extended_DMV is not present in the picture header</td></tr><tr><td>1h</td><td></td><td>Extended_DMV is present in the picture header</td></tr></tbody></table>	Value	Name	Description	0h	[Default]	Extended_DMV is not present in the picture header	1h		Extended_DMV is present in the picture header	
Value	Name	Description									
0h	[Default]	Extended_DMV is not present in the picture header									
1h		Extended_DMV is present in the picture header									



MFD_VP8_BSD_OBJECT

MFD_VP8_BSD_OBJECT			
Source:	VideoCS		
Length Bias:	2		
<p>The MFD_VP8_BSD_OBJECT command is the only primitive command for the VP8 Decoding Pipeline. The Partitions of the bitstream is loaded as indirect data object.</p> <p>Before issuing a MFD_VP8_BSD_OBJECT command, all VP8 frame level 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_VP8_BSD_OBJECT command.</p> <p>Context switch interrupt is not supported by this command.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFD_VP8_BSD_OBJECT
		Format:	OpCode
	26:24	Media Command OpCode	
		Default Value:	4h VP8_DEC
		Format:	OpCode
	23:21	subOpcodeA	
Default Value:		1h	
Format:		OpCode	
20:16	subOpcodeB		
	Default Value:	8h	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	14h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:21	Reserved	
		Format:	MBZ
	20:16	Partition0 CPBAC Entropy Count Pass the Partition0 CPBAC State to HW. Max value is 24.	



MFD_VP8_BSD_OBJECT				
	15:8	Partition0 CPBAC Entropy Range Pass the Partition0 CPBAC State to HW.		
	7:6	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
	5:4	Coded Num of Coeff Token Partitions Num of Partitions = $2^{\text{CodedNumCoeffTokenPartitions}}$. 0 = 1 Partition only 1 = 2 Partitions 2 = 4 Partitions 3 = 8 Partitions are present in the bitstream.		
	3	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
2:0	Partition0 FirstMBBitOffset from Frame Header Allow HW to jump to the location in the bitstream where per MB information starts in the Partition0.			
2	31:24	Partition0 CPBAC Entropy Value Pass the Partition0 CPBAC State to HW.		
	23:0	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
	MBZ			
3	31:24	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
23:0	Indirect Partition0 Data Length This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Partition Start Offset field is ignored. The Partition is byte aligned in both ends. It is the length in bytes of the bitstream data for the current partition. It includes the first byte of the first macroblock and the last byte of the last macroblock in the partition. <table border="1" style="width: 100%; text-align: center;"><tr><td>Programming Notes</td></tr><tr><td>This needs to be set to the (actual Partition 0 length + 1) in bytes</td></tr></table>	Programming Notes	This needs to be set to the (actual Partition 0 length + 1) in bytes	
Programming Notes				
This needs to be set to the (actual Partition 0 length + 1) in bytes				
4	31:0	Indirect Partition0 Data Start Offset This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This offset 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 VP8 bitstream data in each partition.		
5	31:24	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
23:0	Indirect Partition1 Data Length This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Partition Start Offset field is ignored. The Partition is byte aligned in both ends.			



MFD_VP8_BSD_OBJECT						
		<p>It is the length in bytes of the bitstream data for the current partition. It includes the first byte of the first macroblock and the last byte of the last macroblock in the partition.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">This needs to be set to the (actual Partition 1 length + 1) in bytes</td> </tr> </table>	Programming Notes		This needs to be set to the (actual Partition 1 length + 1) in bytes	
Programming Notes						
This needs to be set to the (actual Partition 1 length + 1) in bytes						
6	31:0	<p>Indirect Partition1 Data Start Offset This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This offset 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 VP8 bitstream data in each partition.</p>				
7	31:24	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					
	23:0	<p>Indirect Partition2 Data Length This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Partition Start Offset field is ignored. The Partition is byte aligned in both ends. It is the length in bytes of the bitstream data for the current partition. It includes the first byte of the first macroblock and the last byte of the last macroblock in the partition.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">This needs to be set to the (actual Partition 2 length + 1) in bytes</td> </tr> </table>	Programming Notes		This needs to be set to the (actual Partition 2 length + 1) in bytes	
Programming Notes						
This needs to be set to the (actual Partition 2 length + 1) in bytes						
8	31:0	<p>Indirect Partition2 Data Start Offset This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This offset 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 VP8 bitstream data in each partition.</p>				
9	31:24	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					
	23:0	<p>Indirect Partition3 Data Length This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Partition Start Offset field is ignored. The Partition is byte aligned in both ends. It is the length in bytes of the bitstream data for the current partition. It includes the first byte of the first macroblock and the last byte of the last macroblock in the partition.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">This needs to be set to the (actual Partition 3 length + 1) in bytes</td> </tr> </table>	Programming Notes		This needs to be set to the (actual Partition 3 length + 1) in bytes	
Programming Notes						
This needs to be set to the (actual Partition 3 length + 1) in bytes						
10	31:0	<p>Indirect Partition3 Data Start Offset This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This offset 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 VP8 bitstream data in each partition.</p>				



MFD_VP8_BSD_OBJECT				
11	31:24	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
23:0	<p>Indirect Partition4 Data Length</p> <p>This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Partition Start Offset field is ignored. The Partition is byte aligned in both ends.</p> <p>It is the length in bytes of the bitstream data for the current partition. It includes the first byte of the first macroblock and the last byte of the last macroblock in the partition.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This needs to be set to the (actual Partition 4 length + 1) in bytes</p>	Programming Notes		
Programming Notes				
12	31:0	<p>Indirect Partition4 Data Start Offset</p> <p>This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This offset is relative to the MFD Indirect Object Base Address.</p> <p>Hardware ignores this field if indirect data is not present. It is a byte-aligned address for the VP8 bitstream data in each partition.</p>		
13	31:24	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
23:0	<p>Indirect Partition5 Data Length</p> <p>This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Partition Start Offset field is ignored. The Partition is byte aligned in both ends.</p> <p>It is the length in bytes of the bitstream data for the current partition. It includes the first byte of the first macroblock and the last byte of the last macroblock in the partition.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This needs to be set to the (actual Partition 5 length + 1) in bytes</p>	Programming Notes		
Programming Notes				
14	31:0	<p>Indirect Partition5 Data Start Offset</p> <p>This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This offset is relative to the MFD Indirect Object Base Address.</p> <p>Hardware ignores this field if indirect data is not present. It is a byte-aligned address for the VP8 bitstream data in each partition.</p>		
15	31:24	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
23:0	<p>Indirect Partition6 Data Length</p> <p>This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Partition Start Offset field is ignored. The Partition is byte aligned in both ends.</p> <p>It is the length in bytes of the bitstream data for the current partition. It includes the first byte of the first macroblock and the last byte of the last macroblock in the partition.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This needs to be set to the (actual Partition 6 length + 1) in bytes</p>	Programming Notes		
Programming Notes				



MFD_VP8_BSD_OBJECT											
16	31:0	<p>Indirect Partition6 Data Start Offset</p> <p>This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This offset is relative to the MFD Indirect Object Base Address.</p> <p>Hardware ignores this field if indirect data is not present. It is a byte-aligned address for the VP8 bitstream data in each partition.</p>									
17	31:24	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
23:0	<p>Indirect Partition7 Data Length</p> <p>This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Partition Start Offset field is ignored. The Partition is byte aligned in both ends.</p> <p>It is the length in bytes of the bitstream data for the current partition. It includes the first byte of the first macroblock and the last byte of the last macroblock in the partition.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">This needs to be set to the (actual Partition 7 length + 1) in bytes</td> </tr> </table>	Programming Notes		This needs to be set to the (actual Partition 7 length + 1) in bytes							
Programming Notes											
This needs to be set to the (actual Partition 7 length + 1) in bytes											
18	31:0	<p>Indirect Partition7 Data Start Offset</p> <p>This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This offset is relative to the MFD Indirect Object Base Address.</p> <p>Hardware ignores this field if indirect data is not present. It is a byte-aligned address for the VP8 bitstream data in each partition.</p>									
19	31:24	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
23:0	<p>Indirect Partition8 Data Length</p> <p>This field provides the length in bytes of the indirect data. A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect Partition Start Offset field is ignored. The Partition is byte aligned in both ends.</p> <p>It is the length in bytes of the bitstream data for the current partition. It includes the first byte of the first macroblock and the last byte of the last macroblock in the partition.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">This needs to be set to the (actual Partition 8 length + 1) in bytes</td> </tr> </table>	Programming Notes		This needs to be set to the (actual Partition 8 length + 1) in bytes							
Programming Notes											
This needs to be set to the (actual Partition 8 length + 1) in bytes											
20	31:0	<p>Indirect Partition8 Data Start Offset</p> <p>This field specifies the Graphics Memory starting address of the data to be fetched into BSD Unit for processing. This offset is relative to the MFD Indirect Object Base Address.</p> <p>Hardware ignores this field if indirect data is not present. It is a byte-aligned address for the VP8 bitstream data in each partition.</p>									
21	31	<p>Concealment Method</p> <p>This field specifies the method used for concealment when error is detected.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Intra 16x16 Prediction</td> <td>A copy from the current picture is performed using Intra 16x16 Prediction method.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Inter P Copy</td> <td>A copy from collocated macroblock location is performed from the</td> </tr> </tbody> </table>	Value	Name	Description	0	Intra 16x16 Prediction	A copy from the current picture is performed using Intra 16x16 Prediction method.	1	Inter P Copy	A copy from collocated macroblock location is performed from the
Value	Name	Description									
0	Intra 16x16 Prediction	A copy from the current picture is performed using Intra 16x16 Prediction method.									
1	Inter P Copy	A copy from collocated macroblock location is performed from the									



MFD_VP8_BSD_OBJECT		
		concealment reference indicated by the ConCeal_Pic_Id field.
30:18	Reserved	
	Format:	MBZ
17:16	Conceal_Pic_Id (Concealment Picture ID)	
	Exists If:	[Concealment Method] == 1
	<p>This field identifies the picture in the reference list to be used for concealment. This field is only valid if Concealment Method is Inter P Copy.</p> <ul style="list-style-type: none"> 00 - Last Decoded Picture 01 - Golden Reference Picture 02 - Alternate Reference Picture 03 - User provided Reference Picture 	
15	Reserved	
	Format:	MBZ
14	BSDPrematureComplete Error Handling	
	It occurs in situation where the decode is completed but there are still data in the bitstream.	
	Value	Name
	0	Ignore the error and continue (masked the interrupt), assume the hardware automatically perform the error handling
	1	Set the interrupt to the driver (provide MMIO registers for MB address R/W)??
13	Reserved	
	Format:	MBZ
12	MPR Error (MV out of range) Handling	
	Value	Name
	0	Ignore the error and continue (masked the interrupt), assume the hardware automatically perform the error handling
	1	Set the interrupt to the driver (provide MMIO registers for MB address R/W)??
11	Reserved	
	Format:	MBZ
10	Entropy Error Handling	
	Value	Name
	0	Ignore the error and continue (masked the interrupt), assume the hardware automatically perform the error handling
	1	Set the interrupt to the driver (provide MMIO registers for MB address R/W)??
9	Reserved	
	Format:	MBZ



MFD_VP8_BSD_OBJECT		
8	MB Header Error Handling	
	Value	Name
	0	Ignore the error and continue (masked the interrupt), assume the hardware automatically perform the error handling
	1	Set the interrupt to the driver (provide MMIO registers for MB address R/W)??
7:0	Reserved	
	Format:	MBZ



MFX_AVC_DIRECTMODE_STATE

MFX_AVC_DIRECTMODE_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>This is a picture level command and is issued once per picture. All DMV buffers are treated as standard media surfaces, in which the lower 6 bits are used for conveying surface states. Current Pic POC number is assumed to be available in POCList[32 and 33] of the MFX_AVC_DIRECTMODE_STATE Command. This command is only valid in the AVC decoding in VLD and IT modes, and AVC encoder mode. The same command supports both Long and Short DXVA2 AVC Interface. The DMV buffers are not required to be programmed for encoder mode.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_SINGLE_DW
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	1h AVC_COMMON
		Format:	OpCode
	23:21	SubOpcodeA	
Default Value:		0h	
Format:		OpCode	
20:16	SubOpcodeB		
	Default Value:	2h	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	0045h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1..2	63:6	Direct MV Buffer Base Address for Picture 0 - Read/Write	
		Format:	GraphicsAddress63-6
	<p>Note: This field is changed to one per frame (both top and bottom field share the same Direct MV Buffer Base Address).</p>		
	<p>This field provides the 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</p>		



MFV_AVC_DIRECTMODE_STATE													
	<p>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 557,056 bytes for 1 frame. 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) It is only valid if the current picture is a progressive frame, MbAff frame, or a top field. There are a total of 32 reference picture (previously decoded) Direct MV Buffers (0 to 31, not including the DMV write buffer 32 and 33 of the current picture) to read in 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/Bottom 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]].</p>												
5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
3..32	<p>63:6 Direct MV Buffer Base Address for Reference Frame 1 to 15 (In Frame)</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-6</td> </tr> </table> <p>This field is for the Pre-Deblocking Destination Address, and provides the base address of the DMV buffer for reference frames 2 to 31. They are needed if the current B-Picture has MBs coded in direct mode. This 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 current write buffer of the current picture) to read in the corresponding DMV. Each read 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). The adjacent DMV buffers are paired ([2 and 3], [4 and 5], [N and N+1], ..[30 and 31]). This field is changed to one per frame: both top and bottom field share the same Direct MV Buffer Base Address.</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Programming Notes</td> </tr> <tr> <td>This field is ignored if PreDeblockOutEnable is set to 0 (disable).</td> </tr> </table>	Format:	GraphicsAddress63-6	Programming Notes	This field is ignored if PreDeblockOutEnable is set to 0 (disable).								
Format:	GraphicsAddress63-6												
Programming Notes													
This field is ignored if PreDeblockOutEnable is set to 0 (disable).													
5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
33	<p>31:15 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
	<p>14:13 Direct MV Buffer Base Address for Reference Frame - Tiled Resource Mode For Media Surfaces:</p> <p>This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources
Value	Name	Description											
0h	TRMODE_NONE	No tiled resource											
1h	TRMODE_TILEYF	4KB tiled resources											
2h	TRMODE_TILEYS	64KB tiled resources											



MFX_AVC_DIRECTMODE_STATE		
	3h	Reserved
12:11	Reserved	
	Format:	MBZ
10	Direct MV Buffer Base Address for Reference Frame - Memory Compression Mode Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.	
	Value	Name
	0	Horizontal Compression Mode
	1	Vertical Compression Mode
9	Direct MV Buffer Base Address for Reference Frame - Memory Compression Enable	
	Value	Name
	0	Compression Disable
	Programming Notes	
	This surface is linear surface. This bit must be set to "0" since only TileY/TileYf/TileYs surface is allowed to be compressed	
8:7	Direct MV Buffer Base Address for Reference Frame - Arbitration Priority Control	
	Format:	U2
	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
6:1	Direct MV Buffer Base Address for Reference Frame - Index to Memory Object Control State (MOCS) Tables	
	Format:	U6
	The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.	
	The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
0	Reserved	



MFV_AVC_DIRECTMODE_STATE																	
34..35	63:6	<p>Direct MV Buffer Base Address for Write (Write-Only Buffer)(in frame)</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress63-6</td> </tr> </table> <p>This 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.</p> <p>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).</p> <p>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.</p> <p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and MBZ. Some GraphicsAddress fields only specify the upper address bits. For example, GraphicsAddress[47:12] is a 4KB page address.</p>	Format:	GraphicsAddress63-6													
	Format:	GraphicsAddress63-6															
5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
36	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	<p>Direct MV Buffer Base Address for Write - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
0h	TRMODE_NONE	No tiled resource															
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
12:11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
10	<p>Direct MV Buffer Base Address for Write - Memory Compression Mode Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td>1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode										
Value	Name																
0	Horizontal Compression Mode																
1	Vertical Compression Mode																



MFX_AVC_DIRECTMODE_STATE											
	9	<p>Direct MV Buffer Base Address for Write - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This surface is linear surface. This bit must be set to "0" since only TileY/TileYf/TileYs surface is allowed to be compressed</p>	Value	Name	0	Compression Disable					
	Value	Name									
	0	Compression Disable									
	8:7	<p>Direct MV Buffer Base Address for Write - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b
Value	Name										
00b	Highest priority										
01b	Second highest priority										
10b	Third highest priority										
11b	Lowest priority										
6:1	<p>Direct MV Buffer Base Address for Write - Index to Memory Object Control State (MOCS) Tables</p> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>	Format:	U6								
Format:	U6										
0	Reserved										
37..70	31:0 <p>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 Current Frames/Fields There 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 the frame_store_ID[4:0], which is obtained from RefPicList L0/L1[RefPicIdx]. frame_Store_IDbit[0] (indicator for Top/Bottom 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.</p>										



MFX_AVC_IMG_STATE

MFX_AVC_IMG_STATE		
Source:	VideoCS	
Length Bias:	2	
This must be the very first command to issue after the surface state, the pipe select and base address setting commands. This command supports both Long and Short VLD and IT DXVA2 AVC Decoding Interface.		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
		Format: OpCode
	28:27	Pipeline
		Default Value: 2h MFX_AVC_IMG_STATE
		Format: OpCode
	26:24	Media Command Opcode
		Default Value: 1h AVC_COMMON
Format: OpCode		
23:21	SubOpcode A	
	Default Value: 0h	
	Format: OpCode	
20:16	SubOpcode B	
	Default Value: 0h	
	Format: OpCode	
15:12	Reserved	
	Format: MBZ	
11:0	DWord Length	
	Default Value: 0Ch Excludes DWord (0,1)	
	Format: =n 00Eh, used for normal decode and encode mode000h, a special case to provide a dummy image state for stitch mode operation. In this case, fields in DW1 which is part of the dummy image state command are ignored by hardware.	
1	31:16	Reserved
		Format: MBZ



MFX_AVC_IMG_STATE										
	15:0	<p>Frame Size</p> <table border="1"> <tr> <td>Format:</td> <td>U16 in MB unit</td> </tr> </table> <p>This field is only valid for encoder. Its ignored for decoder. The value for FrameSizeInMBs must match the product of FrameWidthInMBs and FrameHeightInMBs. For encode mode if frame size is 4Kx4K, then we need to specify 65535 in this field as we dont have enough bits to specify 65536. This is used only for cabac zero work insertion when in encode mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[1,65535]</td> <td></td> <td>representing Number of MBs [1,65535]</td> </tr> </tbody> </table>	Format:	U16 in MB unit	Value	Name	Description	[1,65535]		representing Number of MBs [1,65535]
	Format:	U16 in MB unit								
Value	Name	Description								
[1,65535]		representing Number of MBs [1,65535]								
2	31:24	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>(bit[31:24] must be zero to match the DXVA 16-bit definition for FrameHeightInMBsMinus1)</p>	Format:	MBZ						
	Format:	MBZ								
	23:16	<p>Frame Height</p> <table border="1"> <tr> <td>Format:</td> <td>U8-1 in MB unit</td> </tr> </table> <p>It is set to the value of (FrameHeightInMBsMinus1+ 1). Since the max value for FrameHeightInMBs is 255, the max allowed value for FrameHeightInMBsMinus1 is only 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 (1080 divided by 16, and rounded up, i.e. effectively specified as 1088 instead). It is derived from FrameHeightInMbs = (2 - frame_mbs_only_flag) * PicHeightInMapUnits and PicHeightInMbs = FrameHeightInMbs / (1 + field_pic_flag) internally done. For MBAFF, PicHeightInMapUnits is in MB pair unit, so the bitstream sends only half frame height.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,255]</td> <td></td> <td>representing height [1,256]</td> </tr> </tbody> </table>	Format:	U8-1 in MB unit	Value	Name	Description	[0,255]		representing height [1,256]
	Format:	U8-1 in MB unit								
Value	Name	Description								
[0,255]		representing height [1,256]								
15:8	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>(bit[15:8] must be zero to match the DXVA 16-bit definition for FrameWidthInMBsMinus1)</p>	Format:	MBZ							
Format:	MBZ									
7:0	<p>Frame Width</p> <table border="1"> <tr> <td>Format:</td> <td>U8-1 in MB unit</td> </tr> </table> <p>It is set to the value of (FrameWidthInMBsMinus1+ 1). Since the max value for FrameWidthInMBs is 255, the max allowed value for FrameWidthInMBsMinus1 is only 254. The min value for FrameWidthInMBs is 1. Although the max. value that can be specified for FrameWidthInMBs is 255 (in the current implementation), FrameWidthInMBs * FrameWidthInMBs must not exceed the max value of FrameSizeInMBs[14:0]. e.g. for 1920x1080, FrameHeightInMBs[7:0] is equal to 68 (1080 divided by 16, and rounded up, i.e. effectively specified as 1088 instead). It is derived from FrameWidthInMbs = (2 - frame_mbs_only_flag) * PicWidthInMapUnits and</p>	Format:	U8-1 in MB unit							
Format:	U8-1 in MB unit									



MFX_AVC_IMG_STATE																		
		<p>PicWidthInMbs = FrameWidthInMbs / (1 + field_pic_flag) internally done. For MBAFF, PicWidthInMapUnits is in MB pair unit, so the bitstream sends only half frame width.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,255]</td> <td></td> <td>representing width [1,256]</td> </tr> </tbody> </table>	Value	Name	Description	[0,255]		representing width [1,256]										
Value	Name	Description																
[0,255]		representing width [1,256]																
3	31:29	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>(bit[31:29] must be zero to match the DXVA2 8-bit definition for InitQpChroma[1])</p>	Format:	MBZ														
	Format:	MBZ																
	28:24	<p>Second Chroma QP Offset</p> <p>Signed integer value. It should be in the range of -12 to +12 (according to AVC spec).It specifies the offset for determining QP Cr from QP Y. It is set to the upper 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</p>																
	23:21	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>(bit[23:21] must be zero to match the DXVA2 8-bit definition for InitQpChroma[1])</p>	Format:	MBZ														
	Format:	MBZ																
	20:16	<p>First Chroma QP Offset</p> <p>Signed integer value. It should be in the range of -12 to +12 (according to AVC spec).It 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</p>																
	15:14	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																	
13	<p>RhoDomain Rate Control Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This field indicates if RhoDomain related parameters are present in the MFX_AVC_IMAGE_STATE. (AverageMacroblockQP). It enables the Rho Domain statistics collection.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>RhoDomain rate control parameters are not present in MFX_AVC_IMAGE_STATE</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>RhoDomain rate control parameters are present in MFX_AVC_IMAGE_STATE.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="3" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="3">This field must set to '0' for B pictures.</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0	Disable	RhoDomain rate control parameters are not present in MFX_AVC_IMAGE_STATE	1	Enable	RhoDomain rate control parameters are present in MFX_AVC_IMAGE_STATE.	Programming Notes			This field must set to '0' for B pictures.		
Format:	Enable																	
Value	Name	Description																
0	Disable	RhoDomain rate control parameters are not present in MFX_AVC_IMAGE_STATE																
1	Enable	RhoDomain rate control parameters are present in MFX_AVC_IMAGE_STATE.																
Programming Notes																		
This field must set to '0' for B pictures.																		
12	<p>Weighted_Pred_Flag</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>(This field is defined differently from Gen6, Gen7 follows strictly DXVA2 AVC interface.)</p>	Format:	Enable															
Format:	Enable																	



MFX_AVC_IMG_STATE																	
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable [Default]</td> <td>specifies that weighted prediction is not used for P and SP slices</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>specifies that weighted prediction is used for P and SP slices</td> </tr> </tbody> </table>	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						
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															
		<p style="text-align: center;">Programming Notes</p> <p>This field must set to '0' for B and I pictures.</p>															
	11:10	<p>Weighted_BiPred_Idx (This field follows strictly DXVA2 AVC interface.)</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>DEFAULT [Default]</td> <td>Specifies that the default weighted prediction is used for B slices</td> </tr> <tr> <td>1</td> <td>EXPLICIT</td> <td>Specifies that explicit weighted prediction is used for B slices</td> </tr> <tr> <td>2</td> <td>IMPLICIT</td> <td>Specifies that implicit weighted prediction is used for B slices.</td> </tr> <tr> <td>3</td> <td>Reserved</td> <td>Illegal value</td> </tr> </tbody> </table>	Value	Name	Description	0	DEFAULT [Default]	Specifies that the default weighted prediction is used for B 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
Value	Name	Description															
0	DEFAULT [Default]	Specifies that the default weighted prediction is used for B 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															
		<p style="text-align: center;">Programming Notes</p> <p>This field must set to 0 for P and I pictures.</p>															
	9:8	<p>ImgStruct - Image Structure, img_structure[1:0] The current encoding picture structure can only takes on 3 possible values</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Frame Picture</td> </tr> <tr> <td>01b</td> <td>Top Field Picture</td> </tr> <tr> <td>11b</td> <td>Bottom Field Picture</td> </tr> <tr> <td>10b</td> <td>Invalid, not allowed.</td> </tr> </tbody> </table>	Value	Name	00b	Frame Picture	01b	Top Field Picture	11b	Bottom Field Picture	10b	Invalid, not allowed.					
Value	Name																
00b	Frame Picture																
01b	Top Field Picture																
11b	Bottom Field Picture																
10b	Invalid, not allowed.																
		<p style="text-align: center;">Programming Notes</p> <p>img_structure[0] can be used as a flag to distinguish between frame and field structure. It must be consistent with the field_pic_flag setting in the Slice Header. This parameter is specified for Intel interface only, not present in the DXVA as a separate state (instead the img_structure[1] is embedded inside the DXVA picture definition).</p>															
	7:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
4	31:16	<p>MinFrameWSize</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>U16</td> </tr> </table>	Default Value:	0h	Format:	U16											
Default Value:	0h																
Format:	U16																



MFX_AVC_IMG_STATE													
		<p>Minimum Frame Size [15:0] (in Word, 16-bit)(Encoder Only) Minimum Frame Size is specified to compensate for intel Rate Control Currently zero fill (no need to perform emulation byte insertion) is done only to the end of the CABAC_ZERO_WORD insertion (if any) at the last slice of a picture. Intel encoder parameter, not part of DXVA. The caller should always make sure that the value, represented by Minimum Frame Size, is always less than maximum frame size FrameBitRateMax (DWORD 10 bits 29:16). This field is reserved in Decode mode.</p> <p>The programmable range is $0 \dots 2^{18}-1$ When MinFrameWSizeUnits is 00. Programmable range is $0 \dots 2^{20}-1$ when MinFrameWSizeUnits is 01. Programmable range is $0 \dots 2^{26}-1$ when MinFrameWSizeUnits is 10. Programmable range is $0 \dots 2^{32}-1$ when MinFrameWSizeUnits is 11.</p>											
15	MbStatEnabled	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Enable reading in MB status buffer (a.k.a. encoding stream-out buffer) Note: For multi-pass encoder, all passes except the first one need to set this value to 1. By setting the first pass to 0, it does save some memory bandwidth.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>Disable Reading of Macroblock Status Buffer</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Enable Reading of Macroblock Status Buffer</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0	Disable	Disable Reading of Macroblock Status Buffer	1	Enable	Enable Reading of Macroblock Status Buffer
Format:	Enable												
Value	Name	Description											
0	Disable	Disable Reading of Macroblock Status Buffer											
1	Enable	Enable Reading of Macroblock Status Buffer											
14	LoadSlicePointerFlag	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>LoadBitStreamPointerPerSlice (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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>Load BitStream Pointer only once for the first slice of a frame</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Load/reload BitStream Pointer only once for the each slice, reload the start location of the bitstream buffer from the Indirect PAK-BSE Object Data Start Address field</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0	Disable	Load BitStream Pointer only once for the first slice of a frame	1	Enable	Load/reload BitStream Pointer only once for the each slice, reload the start location of the bitstream buffer from the Indirect PAK-BSE Object Data Start Address field
Format:	Enable												
Value	Name	Description											
0	Disable	Load BitStream Pointer only once for the first slice of a frame											
1	Enable	Load/reload BitStream Pointer only once for the each slice, reload the start location of the bitstream buffer from the Indirect PAK-BSE Object Data Start Address field											
13	Reserved												
12	MvUnpackedFlag	<p>MVUnPackedEnable (Encoder Only) This field is reserved in Decode mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>PACKED</td> <td>use packed MV format (compliant to DXVA)</td> </tr> <tr> <td>1</td> <td>UNPACKED</td> <td>use unpacked 8MV/32MV format only</td> </tr> </tbody> </table>	Value	Name	Description	0	PACKED	use packed MV format (compliant to DXVA)	1	UNPACKED	use unpacked 8MV/32MV format only		
Value	Name	Description											
0	PACKED	use packed MV format (compliant to DXVA)											
1	UNPACKED	use unpacked 8MV/32MV format only											



MFX_AVC_IMG_STATE																	
11:10	<p>ChromaFormatIdc Chroma Format IDC, ChromaFormatIdc[1:0]It specifies the sampling of chroma component (Cb, Cr) in the current picture as follows :</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>monochrome picture</td> <td>Desc</td> </tr> <tr> <td>01b</td> <td>4:2:0 picture</td> <td>Desc</td> </tr> <tr> <td>10b</td> <td>4:2:2 picture (not supported)</td> <td></td> </tr> <tr> <td>11b</td> <td>4:4:4 picture (not supported)</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>It is set to the value of the syntax element read from the current active SPS.The corresponding Monochrome Flag (monochrome_flag) can be derived from this field.</p>		Value	Name	Description	00b	monochrome picture	Desc	01b	4:2:0 picture	Desc	10b	4:2:2 picture (not supported)		11b	4:4:4 picture (not supported)	
Value	Name	Description															
00b	monochrome picture	Desc															
01b	4:2:0 picture	Desc															
10b	4:2:2 picture (not supported)																
11b	4:4:4 picture (not supported)																
9	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ													
Format:	MBZ																
8	<p>MbMvFormatFlag Use MB level MvFormat flag (Encoder Only)</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>IGNORE</td> <td>HW PAK ignore MvFormat in the MB data. When bit 12 == 0, all MBs use packed MV formatWhen bit 12 == 1, each MB data must use unpacked MV format, 8MV when there is no minor MV involved, and 32MV if there are some minor MVs.</td> </tr> <tr> <td>1</td> <td>FOLLOW</td> <td>HW PAK will follow MvFormat value set within each MB data.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>They must take one of the two values: the 8MV unpacked format (MvFormat =101b), and the 32MV unpacked format (MvFormat =110b).This bit can be set only when MvUnpackedFlag (bit 12 of this register) is set otherwise system could hang.</p>		Value	Name	Description	0	IGNORE	HW PAK ignore MvFormat in the MB data. When bit 12 == 0, all MBs use packed MV formatWhen bit 12 == 1, each MB data must use unpacked MV format, 8MV when there is no minor MV involved, and 32MV if there are some minor MVs.	1	FOLLOW	HW PAK will follow MvFormat value set within each MB data.						
Value	Name	Description															
0	IGNORE	HW PAK ignore MvFormat in the MB data. When bit 12 == 0, all MBs use packed MV formatWhen bit 12 == 1, each MB data must use unpacked MV format, 8MV when there is no minor MV involved, and 32MV if there are some minor MVs.															
1	FOLLOW	HW PAK will follow MvFormat value set within each MB data.															
7	<p>EntropyCodingFlag Entropy Coding Flag, entropy_coding_flag</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAVLC bit-serial encoding mode</td> <td>Desc</td> </tr> <tr> <td>1</td> <td>CABAC bit-serial encoding mode.</td> <td>Desc</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>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.</p>		Value	Name	Description	0	CAVLC bit-serial encoding mode	Desc	1	CABAC bit-serial encoding mode.	Desc						
Value	Name	Description															
0	CAVLC bit-serial encoding mode	Desc															
1	CABAC bit-serial encoding mode.	Desc															



MFX_AVC_IMG_STATE										
6	<p>ImgDisposableFlag Current Img Disposable Flag or Non-Reference Picture Flag</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>REFERENCE</td> <td>the current decoding picture may be used as a reference picture for others</td> </tr> <tr> <td>1</td> <td>DISPOSABLE</td> <td>the current decoding picture is not used as a reference picture (e.g. a B-picture cannot be a reference picture for any subsequent decoding)</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>It is derived from <code>ImgDisposableFlag = (nal_ref_idc == 0)</code>. <code>nal_ref_idc</code> is a syntax element from a NAL unit. When this flag is set, no reference picture and DMV are written out. This field is only valid for VLD decoding mode.</p>	Value	Name	Description	0	REFERENCE	the current decoding picture may be used as a reference picture for others	1	DISPOSABLE	the current decoding picture is not used as a reference picture (e.g. a B-picture cannot be a reference picture for any subsequent decoding)
Value	Name	Description								
0	REFERENCE	the current decoding picture may be used as a reference picture for others								
1	DISPOSABLE	the current decoding picture is not used as a reference picture (e.g. a B-picture cannot be a reference picture for any subsequent decoding)								
5	<p>ConstrainedIPredFlag Constrained Intra Prediction Flag, <code>constrained_ipred_flag</code>It is set to the value of the syntax element in the current active PPS.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>INTRA_AND_INTER</td> <td>allows both intra and inter neighboring MB to be used in the intra-prediction encoding of the current MB.</td> </tr> <tr> <td>1</td> <td>INTRA_ONLY</td> <td>allows only to use neighboring Intra MBs in the intra-prediction encoding of the current MB. If the neighbor is an inter MB, it is considered as not available.</td> </tr> </tbody> </table>	Value	Name	Description	0	INTRA_AND_INTER	allows both intra and inter neighboring MB to be used in the intra-prediction encoding of the current MB.	1	INTRA_ONLY	allows only to use neighboring Intra MBs in the intra-prediction encoding of the current MB. If the neighbor is an inter MB, it is considered as not available.
Value	Name	Description								
0	INTRA_AND_INTER	allows both intra and inter neighboring MB to be used in the intra-prediction encoding of the current MB.								
1	INTRA_ONLY	allows only to use neighboring Intra MBs in the intra-prediction encoding of the current MB. If the neighbor is an inter MB, it is considered as not available.								
4	<p>Direct8x8InfFlag Direct 8x8 Inference Flag, <code>direct_8x8_inference_flag</code>It 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 <code>frame_mbs_only_flag</code> is equal to 0, <code>direct_8x8_inference_flag</code> shall be equal to 1. It must be consistent with the <code>frame_mbs_only_flag</code> and <code>transform_8x8_mode_flag</code> settings.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SUBBLOCK</td> <td>allows subpartitioning to go below 8x8 block size (i.e. 4x4, 8x4 or 4x8)</td> </tr> <tr> <td>1</td> <td>BLOCK</td> <td>allows processing only at 8x8 block size. MB Info is stored for 8x8 block size.</td> </tr> </tbody> </table>	Value	Name	Description	0	SUBBLOCK	allows subpartitioning to go below 8x8 block size (i.e. 4x4, 8x4 or 4x8)	1	BLOCK	allows processing only at 8x8 block size. MB Info is stored for 8x8 block size.
Value	Name	Description								
0	SUBBLOCK	allows subpartitioning to go below 8x8 block size (i.e. 4x4, 8x4 or 4x8)								
1	BLOCK	allows processing only at 8x8 block size. MB Info is stored for 8x8 block size.								
3	<p>Transform8x8Flag 8x8 IDCT Transform Mode Flag, <code>trans8x8_mode_flag</code>Specifies 8x8 IDCT transform may be used in this pictureIt is set to the value of the syntax element in the current active PPS.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>4x4</td> <td>no 8x8 IDCT Transform, only 4x4 IDCT transform blocks are present</td> </tr> <tr> <td>1</td> <td>8x8</td> <td>8x8 Transform is allowed</td> </tr> </tbody> </table>	Value	Name	Description	0	4x4	no 8x8 IDCT Transform, only 4x4 IDCT transform blocks are present	1	8x8	8x8 Transform is allowed
Value	Name	Description								
0	4x4	no 8x8 IDCT Transform, only 4x4 IDCT transform blocks are present								
1	8x8	8x8 Transform is allowed								



MFX_AVC_IMG_STATE											
	2	<p>FrameMbOnlyFlag Frame MB only flag, frame_mbs_only_flag.It is set to the value of the syntax element in the current active SPS.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FALSE</td> <td>not true ; effectively enables the possibility of MBAFF mode.</td> </tr> <tr> <td>1</td> <td>TRUE</td> <td>true, only frame MBs can occur in this sequence, hence disallows the MBAFF mode and field picture.</td> </tr> </tbody> </table>	Value	Name	Description	0	FALSE	not true ; effectively enables the possibility of MBAFF mode.	1	TRUE	true, only frame MBs can occur in this sequence, hence disallows the MBAFF mode and field picture.
	Value	Name	Description								
	0	FALSE	not true ; effectively enables the possibility of MBAFF mode.								
1	TRUE	true, only frame MBs can occur in this sequence, hence disallows the MBAFF mode and field picture.									
1	<p>MbaffFlameFlag MBAFF mode is active, mbaff_frame_flag.It is derived from MbaffFrameFlag = (mb_adaptive_frame_field_flag && ! field_pic_flag). mb_adaptive_frame_field_flag is a syntax element in the current active SPS and field_pic_flag is a syntax element in the current Slice Header. They both are present only if frame_mbs_only_flag is 0. Although mbaff_frame_flag is a Slice Header parameter, its value is expected to be the same for all the slices of a picture.It must be consistent with the mb_adaptive_frame_field_flag, the field_pic_flag and the frame_mbs_only_flag settings.This bit is valid only when the img_structure[1:0] indicates the current picture is a frame.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FALSE</td> <td>not in MBAFF mode</td> </tr> <tr> <td>1</td> <td>TRUE</td> <td>in MBAFF mode</td> </tr> </tbody> </table>	Value	Name	Description	0	FALSE	not in MBAFF mode	1	TRUE	in MBAFF mode	
Value	Name	Description									
0	FALSE	not in MBAFF mode									
1	TRUE	in MBAFF mode									
0	<p>FieldPicFlag Field picture flag, field_pic_flag, specifies the current slice is a coded field or not.It is set to the same value as the syntax element in the Slice Header. It must be consistent with the img_structure[1:0] and the frame_mbs_only_flag settings.Although field_pic_flag is a Slice Header parameter, its value is expected to be the same for all the slices of a picture.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>FRAME</td> <td>a slice of a coded frame</td> </tr> <tr> <td>1h</td> <td>FIELD</td> <td>a slice of a coded field</td> </tr> </tbody> </table>	Value	Name	Description	0h	FRAME	a slice of a coded frame	1h	FIELD	a slice of a coded field	
Value	Name	Description									
0h	FRAME	a slice of a coded frame									
1h	FIELD	a slice of a coded field									
5 [ExistsIf]Encode Only	31	<p>Trellis Quantization Enabled (TQEnb) Format: Enable</p> <p>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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Use Normal</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Use Trellis quantization</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	Use Normal	1h	Enable	Use Trellis quantization
Value	Name	Description									
0h	Disable	Use Normal									
1h	Enable	Use Trellis quantization									



MFX_AVC_IMG_STATE																										
30:28	<p>Trellis Quantization Rounding (TQR) This rounding scheme is only applied to the quantized coefficients ranging from 0 to 1 when TQEnb is set to 1 in AVC CABAC mode. One of the following values is added to quantized coefficients before truncating fractional part.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td></td> <td>Add 1/8</td> </tr> <tr> <td>001b</td> <td></td> <td>Add 2/8</td> </tr> <tr> <td>010b</td> <td></td> <td>Add 3/8</td> </tr> <tr> <td>011b</td> <td></td> <td>Add 4/8 (rounding 0.5)</td> </tr> <tr> <td>100b</td> <td></td> <td>Add 5/8</td> </tr> <tr> <td>101b</td> <td></td> <td>Add 6/8</td> </tr> <tr> <td>110b</td> <td>Default</td> <td>Add 7/8 (Default rounding 0.875)</td> </tr> </tbody> </table>		Value	Name	Description	000b		Add 1/8	001b		Add 2/8	010b		Add 3/8	011b		Add 4/8 (rounding 0.5)	100b		Add 5/8	101b		Add 6/8	110b	Default	Add 7/8 (Default rounding 0.875)
Value	Name	Description																								
000b		Add 1/8																								
001b		Add 2/8																								
010b		Add 3/8																								
011b		Add 4/8 (rounding 0.5)																								
100b		Add 5/8																								
101b		Add 6/8																								
110b	Default	Add 7/8 (Default rounding 0.875)																								
27	<p>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, TQChromaDisable=1.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Enable Trellis Quantization chroma</td> </tr> <tr> <td>1h</td> <td>Default</td> <td>Disable Trellis Quantization chroma</td> </tr> </tbody> </table>		Value	Name	Description	0h		Enable Trellis Quantization chroma	1h	Default	Disable Trellis Quantization chroma															
Value	Name	Description																								
0h		Enable Trellis Quantization chroma																								
1h	Default	Disable Trellis Quantization chroma																								
26:17	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ																						
Format:	MBZ																									
16	<p>NonFirstPassFlag This signals the current pass is not the first pass. It will imply designate HW behavior: e.g</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Always use the MbQpY from initial PAK inline object for all passes of PAK</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Use MbQpY from stream-out buffer if MbRateCtrlFlag is set to 1</td> </tr> </tbody> </table>		Value	Name	Description	0h	Disable	Always use the MbQpY from initial PAK inline object for all passes of PAK	1h	Enable	Use MbQpY from stream-out buffer if MbRateCtrlFlag is set to 1															
Value	Name	Description																								
0h	Disable	Always use the MbQpY from initial PAK inline object for all passes of PAK																								
1h	Enable	Use MbQpY from stream-out buffer if MbRateCtrlFlag is set to 1																								
15:13	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ																						
Format:	MBZ																									
12	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ																						
Format:	MBZ																									
11:10	<p>MinFrameWSizeUnits This field is the Minimum Frame Size Units</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>compatibility mode</td> <td>Minimum Frame Size is in old mode (words, 2bytes)</td> </tr> <tr> <td>01b</td> <td>16 byte</td> <td>Minimum Frame Size is in 16bytes</td> </tr> <tr> <td>10b</td> <td>4Kb</td> <td>Minimum Frame Size is in 4Kbytes</td> </tr> </tbody> </table>		Value	Name	Description	00b	compatibility 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												
Value	Name	Description																								
00b	compatibility 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																								



MFX_AVC_IMG_STATE			
	11b	16Kb	Minimum Frame Size is in 16Kbytes
9	MbRateCtrlFlag - MB level Rate Control Enabling Flag MB Rate Control conformance mask		
	Value	Name	Description
	0h	Disable	Apply accumulative delta QP for consecutive passes on top of the macroblock QP values in inline data
	1h	Enable	Apply RC QP delta to suggested QP values in Macroblock Status Buffer except the first pass.
	Programming Notes		
This field is ignored when MacroblockStatEnable is disabled or MB level Rate control flag for the current MB is disable in Macroblock Status Buffer.			
8	Reserved		
	Format:		MBZ
7	Intra/InterMblpcmFlag - ForceIPCMControlMask This field is to Force IPCM for Intra or Inter Macroblock size conformance mask.		
	Value	Name	Description
	0h	Disable	Do not change intra or Inter macroblocks even
	1h	Enable	Change intra or Inter macroblocks MB_type to IPCM
	Programming Notes		
This field is ignored when MacroblockStatEnable is disabled or MB level Intra MB conformance flag for the current MB is disable in Macroblock Status Buffer.			
6:4	Reserved		
	Format:		MBZ
3	FrameSzUnderFlag - FrameBitRateMinReportMask This is a mask bit controlling if the condition of frame level bit count is less than FrameBitRateMin		
	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.
	Programming Notes		
This field is ignored when MacroblockStatEnable is disabled or MB level Intra MB conformance flag for the current MB is disable in Macroblock Status Buffer.			
2	FrameSzOverFlag - FrameBitRateMaxReportMask This is a mask bit controlling if the condition of frame level bit count exceeds FrameBitRateMax.		
	Value	Name	Description
	0	Disable	Do not update bit0 of MFC_IMAGE_STATUS control register.
	Programming Notes		
This field is ignored when MacroblockStatEnable is disabled or MB level Intra MB conformance flag for the current MB is disable in Macroblock Status Buffer.			



MFX_AVC_IMG_STATE											
	1	<table border="1"> <tr> <td>Enable</td> <td>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.</td> </tr> </table>	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.							
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	<p>InterMbMaxBitFlag - InterMBMaxSizeReportMask This is a mask bit controlling if the condition of any inter MB in the frame exceeds InterMBMaxSize.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>Do not update bit0 of MFC_IMAGE_STATUS control register.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>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.</td> </tr> </tbody> </table>	Value	Name	Description	0	Disable	Do not update bit0 of MFC_IMAGE_STATUS control register.	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.
Value	Name	Description									
0	Disable	Do not update bit0 of MFC_IMAGE_STATUS control register.									
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	<p>IntraMbMaxBitFlag - IntraMBMaxSizeReportMask This is a mask bit controlling if the condition of any intra MB in the frame exceeds IntraMBMaxSize.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Do not update bit0 of MFC_IMAGE_STATUS control register.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>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.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	Do not update bit0 of MFC_IMAGE_STATUS control register.	1	Enable	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.
Value	Name	Description									
0h	Disable	Do not update bit0 of MFC_IMAGE_STATUS control register.									
1	Enable	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 [ExistsIf]Encode Only	31:28	Reserved									
	27:16	<p>InterMbMaxSz</p> <table border="1"> <tr> <td>Format:</td> <td>U12</td> </tr> </table> <p>This field, Inter MB Conformance Max size limit, indicates the allowed max bit count size for Inter MB</p>	Format:	U12							
	Format:	U12									
	15:12	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ										
11:0	<p>IntraMbMaxSz</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Intra Only</td> </tr> <tr> <td>Format:</td> <td>U12</td> </tr> </table> <p>This field, Intra MB Conformance Max size limit, indicates the allowed max bit count size for Intra MB</p> <p>All IPCM MBs should ignore this Max size limit.</p>	Exists If:	//Intra Only	Format:	U12						
Exists If:	//Intra Only										
Format:	U12										
7 [ExistsIf]Encode Only	31:17	Reserved									
	16	Reserved									
	15:1	Reserved									



MFX_AVC_IMG_STATE											
	0	<p>VSL Top MB Trans8x8flag</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable [Default]</td> <td>VSL will only fetch the current MB data.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>When this bit is set VSL will make extra fetch to memory to fetch the MB data for top MB.</td> </tr> </tbody> </table>	Value	Name	Description	0	Disable [Default]	VSL will only fetch the current MB data.	1	Enable	When this bit is set VSL will make extra fetch to memory to fetch the MB data for top MB.
Value	Name	Description									
0	Disable [Default]	VSL will only fetch the current MB data.									
1	Enable	When this bit is set VSL will make extra fetch to memory to fetch the MB data for top MB.									
8 [ExistsIf]Encode Only	31:24	<p>SliceDeltaQpMax[3]</p> <table border="1"> <tr> <td>Format:</td> <td>S7</td> </tr> <tr> <td colspan="2">Range: [0:MAX_QP_DELTA]</td> </tr> <tr> <td colspan="2">This field is the Slice level delta QP for total bit-count above FrameBitRateMax - first 1/8 region 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 exceeds FrameBitRateMax but is within 1/8 of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of (FrameBitRateMax, (FrameBitRateMax+ FrameBitRateMaxDelta»3).</td> </tr> </table>	Format:	S7	Range: [0:MAX_QP_DELTA]		This field is the Slice level delta QP for total bit-count above FrameBitRateMax - first 1/8 region 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 exceeds FrameBitRateMax but is within 1/8 of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of (FrameBitRateMax, (FrameBitRateMax+ FrameBitRateMaxDelta»3).				
	Format:	S7									
	Range: [0:MAX_QP_DELTA]										
	This field is the Slice level delta QP for total bit-count above FrameBitRateMax - first 1/8 region 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 exceeds FrameBitRateMax but is within 1/8 of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of (FrameBitRateMax, (FrameBitRateMax+ FrameBitRateMaxDelta»3).										
23:16	<p>SliceDeltaQpMax[2]</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> <tr> <td colspan="2">Range: [0:MAX_QP_DELTA]</td> </tr> <tr> <td colspan="2">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).</td> </tr> </table>	Format:	U8	Range: [0:MAX_QP_DELTA]		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).					
Format:	U8										
Range: [0:MAX_QP_DELTA]											
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	<p>SliceDeltaQpMax[1]</p> <table border="1"> <tr> <td>Format:</td> <td>S7</td> </tr> <tr> <td colspan="2">Range: [0:MAX_QP_DELTA]</td> </tr> <tr> <td colspan="2">This field is the Slice level delta QP for bit-count above FrameBitRateMax - above 1/4 and below 1/2 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 1/4 and 1/2 of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta»2), (FrameBitRateMax+ FrameBitRateMaxDelta»1).</td> </tr> </table>	Format:	S7	Range: [0:MAX_QP_DELTA]		This field is the Slice level delta QP for bit-count above FrameBitRateMax - above 1/4 and below 1/2 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 1/4 and 1/2 of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta»2), (FrameBitRateMax+ FrameBitRateMaxDelta»1).					
Format:	S7										
Range: [0:MAX_QP_DELTA]											
This field is the Slice level delta QP for bit-count above FrameBitRateMax - above 1/4 and below 1/2 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 1/4 and 1/2 of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta»2), (FrameBitRateMax+ FrameBitRateMaxDelta»1).											
7:0	<p>SliceDeltaQpPMax[0]</p> <table border="1"> <tr> <td>Format:</td> <td>S7</td> </tr> <tr> <td colspan="2">Range: [0:MAX_QP_DELTA]</td> </tr> </table>	Format:	S7	Range: [0:MAX_QP_DELTA]							
Format:	S7										
Range: [0:MAX_QP_DELTA]											



MFX_AVC_IMG_STATE			
	<p>This field is the Slice level delta QP for bit-count above FrameBitRateMax - above 1/2 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 above FrameBitRateMax by more than half the distance of FrameBitRateMaxDelta , i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta»1), infinite).</p>		
9 [ExistsIf]Encode Only	31:24 SliceDeltaQpMin[3] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="text-align: center;">S7</td></tr></table> Range: [0:MAX_QP_DELTA] This field is the Slice level delta QP for total bit-count below FrameBitRateMin - first 1/8 region 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 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).		S7
		S7	
	23:16 SliceDeltaQpMin[2] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="text-align: center;">S7</td></tr></table> Range: [0:MAX_QP_DELTA] This field is the Slice level delta QP for bit-count below FrameBitRateMin - below 1/ 8 and above 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 one-eighth and quarter the distance of FrameBitRateMinDelta from FrameBitRateMin, i.e., in the range of ((FrameBitRateMin- FrameBitRateMinDelta»2), (FrameBitRateMin- FrameBitRateMinDelta»3)).		S7
		S7	
15:8 SliceDeltaQpMin[1] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="text-align: center;">S7</td></tr></table> Range: [0:MAX_QP_DELTA] This field is the Slice level delta QP for bit-count below FrameBitRateMin- below 1/4 and above 1/ 2 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 quarter and half the distance of FrameBitRateMinDelta from FrameBitRateMin, i.e., in the range of ((FrameBitRateMin- FrameBitRateMinDelta»1), (FrameBitRateMin- FrameBitRateMinDelta»2)).		S7	
	S7		
7:0 SliceDeltaQpMin[0] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="text-align: center;">S7</td></tr></table> Range: [0:MAX_QP_DELTA] This field is the Slice Level Delta QP for bit-count below FrameBitRateMin - below 1/ 2 This field is used to calculate the suggested slice QP into the MFC_IMAGE_STATUS		S7	
	S7		



MFX_AVC_IMG_STATE												
		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 [ExistsIf]Encode Only	31	FrameBitrateMaxUnit This field is the Frame Bitrate Maximum Limit Units.										
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Byte</td> <td>FrameBitRateMax is in units of 32 Bytes when FrameBitrateMaxUnitMode is 1 and in units of 128 Bytes if FrameBitrateMaxUnitMode is 0</td> </tr> <tr> <td>1</td> <td>Kilo Byte</td> <td>FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0</td> </tr> </tbody> </table>	Value	Name	Description	0	Byte	FrameBitRateMax is in units of 32 Bytes when FrameBitrateMaxUnitMode is 1 and in units of 128 Bytes if FrameBitrateMaxUnitMode is 0	1	Kilo Byte	FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0	
		Value	Name	Description								
	0	Byte	FrameBitRateMax is in units of 32 Bytes when FrameBitrateMaxUnitMode is 1 and in units of 128 Bytes if FrameBitrateMaxUnitMode is 0									
	1	Kilo Byte	FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0									
	30	FrameBitrateMaxUnitMode This field is the Frame Bitrate Maximum Limit Units.										
	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-pass gets triggered (when enabled). In other words, multi-pass is triggered when the actual 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..	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-512KB</td> <td></td> <td>The programmable range is 0-512KB when FrameBitrateMaxUnit is 0.</td> </tr> <tr> <td>0-8190KB</td> <td></td> <td>The programmable range is 0-8190KB when FrameBitrateMaxUnit is 1.</td> </tr> </tbody> </table>	Value	Name	Description	0-512KB		The programmable range is 0-512KB when FrameBitrateMaxUnit is 0.	0-8190KB		The programmable range is 0-8190KB when FrameBitrateMaxUnit is 1.
			Value	Name	Description							
			0-512KB		The programmable range is 0-512KB when FrameBitrateMaxUnit is 0.							
	0-8190KB		The programmable range is 0-8190KB when FrameBitrateMaxUnit is 1.									
15	FrameBitrateMinUnit This field is the Frame Bitrate Minimum Limit Units.	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Byte</td> <td>FrameBitRateMax is in units of 32 Bytes when FrameBitrateMinUnitMode is 1 and in units of 128 Bytes if FrameBitrateMinUnitMode is 0</td> </tr> <tr> <td>1</td> <td>Kilo Byte</td> <td>FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0</td> </tr> </tbody> </table>	Value	Name	Description	0	Byte	FrameBitRateMax is in units of 32 Bytes when FrameBitrateMinUnitMode is 1 and in units of 128 Bytes if FrameBitrateMinUnitMode is 0	1	Kilo Byte	FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0	
		Value	Name	Description								
		0	Byte	FrameBitRateMax is in units of 32 Bytes when FrameBitrateMinUnitMode is 1 and in units of 128 Bytes if FrameBitrateMinUnitMode is 0								
1	Kilo Byte	FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0										



MFX_AVC_IMG_STATE																
	14	<p>FrameBitrateMinUnitMode This field is the Frame Bitrate Minimum Limit Units.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Compatibility mode</td> <td>FrameBitRateMaxUnit is in old mode (128b/16Kb)</td> </tr> <tr> <td>1h</td> <td>New mode</td> <td>FrameBitRateMaxUnit is in new mode (32byte/4Kb)</td> </tr> </tbody> </table>	Value	Name	Description	0h	Compatibility mode	FrameBitRateMaxUnit is in old mode (128b/16Kb)	1h	New mode	FrameBitRateMaxUnit is in new mode (32byte/4Kb)					
	Value	Name	Description													
0h	Compatibility mode	FrameBitRateMaxUnit is in old mode (128b/16Kb)														
1h	New mode	FrameBitRateMaxUnit is in new mode (32byte/4Kb)														
	13:0	<p>FrameBitRateMin RangeThe programmable range 0-512KB When FrameBitrateMinUnit is in 0.Programmable range is 0-8190 KB when FrameBitrateMinUnit is in 1.This field is the Frame Bitrate Minimum Limit ()This field along with FrameBitrateMinUnit determines minimum allowed bits in a Frame before Multi-Pass gets triggered (when enabled). In other words, multi-pass is triggered when the actual frame byte count is less than this value. When FrameBitrateMinUnitMode is 0 (compatibility mode) bits 0:11 should be used, bits 12 and 13 should be 0.</p>														
11 [ExistsIf]Encode Only	31	Slice Stats Streamout Enable														
	30:16	<p>FrameBitRateMaxDelta</p> <table border="1"> <tr> <td>Format:</td> <td>U15</td> </tr> </table> <p>This field is used to select the slice delta QP when FrameBitRateMax Is exceeded. It shares the same FrameBitrateMaxUnit. When FrameBitrateMaxUnitMode is 0(compatibility mode) bits 16:27 should be used, bits 28, 29 and 30 should be 0.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-1024KB</td> <td></td> <td>The Programmable range 0-1024KB when FrameBitRateMaxUnit is 0.</td> </tr> <tr> <td>0-16380KB</td> <td></td> <td>The Programmable range is 0-16380KB when FrameBitRateMaxUnit is 1.</td> </tr> <tr> <td>0h</td> <td>[Default]</td> <td></td> </tr> </tbody> </table>	Format:	U15	Value	Name	Description	0-1024KB		The Programmable range 0-1024KB when FrameBitRateMaxUnit is 0.	0-16380KB		The Programmable range is 0-16380KB when FrameBitRateMaxUnit is 1.	0h	[Default]	
	Format:	U15														
	Value	Name	Description													
	0-1024KB		The Programmable range 0-1024KB when FrameBitRateMaxUnit is 0.													
0-16380KB		The Programmable range is 0-16380KB when FrameBitRateMaxUnit is 1.														
0h	[Default]															
15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ															
14:0	<p>FrameBitRateMinDelta</p> <p>Range: The programmable range 0-1024KB When FrameBitrateMinUnit is in 32Bytes.Programmable range is 0-16380KB when FrameBitrateMinUnit is in 4Kbytes.</p> <p>This field is used to select the slice delta QP when FrameBitRateMin Is exceeded. It shares the same FrameBitrateMinUnit. When FrameBitrateMinUnitMode is 0(compatibility mode) bits 0:11 should be used, bits 12, 13 and 14 should be 0.Note: HW requires the following condition $FrameBitRateMinDelta \leq 2 * FrameBitRateMin$ Must be true, otherwise it may cause unpredicted behavior.</p>															
12	31:21	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ															



MFX_AVC_IMG_STATE											
	20	VMD Error Logic <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 45%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable [Default]</td> <td></td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Error Handling</td> </tr> </tbody> </table>	Value	Name	Description	0	Disable [Default]		1	Enable	Error Handling
	Value	Name	Description								
	0	Disable [Default]									
	1	Enable	Error Handling								
	19	Reserved Format: _____ MBZ									
	18	VAD Error Logic <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Enable [Default]</td> <td>Error reporting ON in case of premature Slice done</td> </tr> <tr> <td>1</td> <td>Disable</td> <td>CABAC Engine will auto decode the bitstream in case of premature slice done.</td> </tr> </tbody> </table>	Value	Name	Description	0	Enable [Default]	Error reporting ON in case of premature Slice done	1	Disable	CABAC Engine will auto decode the bitstream in case of premature slice done.
	Value	Name	Description								
0	Enable [Default]	Error reporting ON in case of premature Slice done									
1	Disable	CABAC Engine will auto decode the bitstream in case of premature slice done.									
17	Reserved										
16	Reserved										
15:0	Reserved Format: _____ MBZ										
13	31:30	Reserved Format: _____ MBZ									
	29	Current Picture Has Performed MMC05 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 reference frames (frames, field pairs, unpaired field) existed in the current DBP for decoding the current picture.									
	23:22	Reserved Format: _____ MBZ									
	21:16	Number of Active Reference Pictures from L1 Format: _____ U6-1 Specifies the initial maximum reference index value minus 1 to access the L1 Reference List. It is extracted from PPS. It corresponds to the number of active reference pictures from L1 to decode the current picture. It can be modified by the slice header if num_ref_idx_active_override_flag is set. Only valid for B picture. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,31]						
	Value	Name									
[0,31]											



MFX_AVC_IMG_STATE						
	15:14	Reserved Format: MBZ				
	13:8	Number of Active Reference Pictures from L0 Format: U6-1 Specifies the initial maximum reference index value minus 1 to access the L0 Reference 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. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,31]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,31]	
	Value	Name				
[0,31]						
7:0	Initial QP Value Format: S7 Range: [-26,25] Initial QP value for a Slice, extracted from PPS. It may further get modified by slice_qp_delta in slice header and mb_qp_delta in MB header.					
14 [ExistsIf] Short Format only	31:24	Log2_max_pic_order_cnt_lsb_minus4 Exists If: //Short Format Only It is a SPS syntax element, used to determine how many bits in the bitstream are used to represent pic_order_cnt_lsb syntax element in the slice header. Unsigned				
	23:16	Log2_max_frame_num_minus4 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 syntax element in the slice header. Unsigned.				
	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_minus1 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 slice group support. Desc				



MFX_AVC_IMG_STATE						
	11	<p>redundant_pic_cnt_present_flag</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> </table> <p>It is a PPS syntax element. Use for Slice Header parsing only, to read-in redundant_pic_cnt, if any, but is not used by H/W, i.e. no support for redundant slice processing.</p>	Exists If:	//Short Format Only		
	Exists If:	//Short Format Only				
	10:8	<p>slice_group_map_type</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> </table> <p>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.</p>	Exists If:	//Short Format Only		
	Exists If:	//Short Format Only				
	7:4	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>IDR flag is decoded from NAL Header Byte</p>	Format:	MBZ		
	Format:	MBZ				
3:2	<p>Pic_order_cnt_type</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> </table> <p>It is a SPS syntax element. Use for Slice Header parsing only.</p>	Exists If:	//Short Format Only			
Exists If:	//Short Format Only					
1	<p>Delta_pic_order_always_zero_flag</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> </table> <p>It is a SPS syntax element. Use for Slice Header parsing only.</p>	Exists If:	//Short Format Only			
Exists If:	//Short Format Only					
0	<p>Pic_order_present_flag</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> </table> <p>It is a PPS syntax element. Use for Slice Header parsing only.</p>	Exists If:	//Short Format Only			
Exists If:	//Short Format Only					
15 [ExistsIf] Short Format only	31:16	<p>Curr Pic Frame Num</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>Derived from Slice Header syntax element</p>	Exists If:	//Short Format Only	Format:	U16
	Exists If:	//Short Format Only				
Format:	U16					
15:0	<p>Slice Group Change Rate</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> <tr> <td>Format:</td> <td>U16-1</td> </tr> </table> <p>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.</p>	Exists If:	//Short Format Only	Format:	U16-1	
Exists If:	//Short Format Only					
Format:	U16-1					



MFX_AVC_IMG_STATE													
16 [ExistsIf]: Short Format only	31	<p>Inter View Order Disable</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> </table> <p>It indicates how to append inter-view picture into initial sorted reference list. (due to ambiguity in the MVC Spec)</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Default [Default]</td> <td>View Order Ascending</td> </tr> <tr> <td>1h</td> <td>Disable</td> <td>View ID Ascending</td> </tr> </tbody> </table>	Exists If:	//Short Format Only	Value	Name	Description	0h	Default [Default]	View Order Ascending	1h	Disable	View ID Ascending
	Exists If:	//Short Format Only											
	Value	Name	Description										
	0h	Default [Default]	View Order Ascending										
	1h	Disable	View ID Ascending										
	30:22	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ											
	21:18	<p>Max View IDX1</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> </table> <p>It is a PPS syntax element corresponding to Anchor/Non-Anchor Reference ListL1 It indicates the maximum number of inter-view picture for Reference List L1</p>	Exists If:	//Short Format Only									
Exists If:	//Short Format Only												
17:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
15:12	<p>Max View IDX0</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> </table> <p>Reference ListL0 It indicates the maximum number of inter-view picture for Reference List L0</p>	Exists If:	//Short Format Only										
Exists If:	//Short Format Only												
11:10	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
9:0	<p>Current Frame View ID</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Short Format Only</td> </tr> </table> <p>It indicates the View ID of the current decoding frame</p>	Exists If:	//Short Format Only										
Exists If:	//Short Format Only												
17	31:22	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ											
	21:16	<p>RhoDomain AverageMacroblockQP</p> <table border="1"> <tr> <td>Exists If:</td> <td>//[RhoDomain Rate Control] == 1</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table>	Exists If:	//[RhoDomain Rate Control] == 1	Format:	U6							
Exists If:	//[RhoDomain Rate Control] == 1												
Format:	U6												
15:9	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												



MFX_AVC_IMG_STATE											
	8	<p>Extended RhoDomain Statistics Enable This parameter enables PAK to generate RhoDomain statistics from marcoblock QP and fractional QP computation.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>RhoDomain statistics generated from Frame Qp and no fractional QP computation.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Enable MB QP based RhoDomain statistics and fractional QP computation.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Reserved: MBZ</p>	Value	Name	Description	0	Disable	RhoDomain statistics generated from Frame Qp and no fractional QP computation.	1	Enable	Enable MB QP based RhoDomain statistics and fractional QP computation.
Value	Name	Description									
0	Disable	RhoDomain statistics generated from Frame Qp and no fractional QP computation.									
1	Enable	Enable MB QP based RhoDomain statistics and fractional QP computation.									
	7:6	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ										
	5:3	<p>Fractional QP offset</p> <table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Start position from the top of the Frame where increased Quantization parameter is added.</p> <p style="text-align: center;">Programming Notes</p> <p>Reserved: MBZ</p>	Format:	U3							
Format:	U3										
	2:0	<p>Fractional QP input In a set of 8 rows, Qp is incremented by 1 for F_qp rows..</p>									
18	31:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ										
19	31:0	<p>Threshold Size in Bytes</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>When a slice exceeds this value in bytes, hardware will end current slice as soon as possible and insert a new slice boundary. Note there is no guarantee that the actual slice size will meet this value, it is a hint to the HW to end the slice as soon as possible (which could be 2-5 macroblocks in the future from this detection point).</p>	Format:	U32							
Format:	U32										
20	31:0	<p>Target Slice Size in Bytes</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table>	Format:	U32							
Format:	U32										



MFX_AVC_REF_IDX_STATE

MFX_AVC_REF_IDX_STATE				
Source:	VideoCS			
Length Bias:	2			
<p>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.</p> <p>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.</p> <p>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.</p>				
Programming Notes				
<p>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 intel 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.</p>				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	3h PARALLEL_VIDEO_PIPE	
		Format:	OpCode	
	28:27	28:27	Pipeline	
			Default Value:	2h MFX_AVC_REF_IDX_STATE
			Format:	OpCode
	26:24	26:24	Command Opcode	
			Default Value:	1h AVC
			Format:	OpCode
	23:21	23:21	SubOpcodeA	
			Default Value:	0h MFX_AVC_REF_IDX_STATE
			Format:	OpCode
20:16	20:16	SubOpcodeB		
		Default Value:	4h MFX_AVC_REF_IDX_STATE	
		Format:	OpCode	



MFX_AVC_REF_IDX_STATE										
	15:12	Reserved Format: MBZ								
	11:0	DWord Length Default Value: 0008h Format: =n Excludes DWords 0,1								
1	31:1	Reserved Format: MBZ								
	0	<p>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 specified for Intel interface only, not present in the DXVA.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>RefPicList 0</td> <td>The list that followed represents RefList L0 (Decoder VLD mode) or Ref Idx Mapping Table L0 (Encoder PAK mode)</td> </tr> <tr> <td>1</td> <td>RefPicList1</td> <td>The list that followed represents RefList L1 (Decoder VLD mode) or Ref Idx Mapping Table L1 (Encoder PAK mode)</td> </tr> </tbody> </table>	Value	Name	Description	0	RefPicList 0	The list that followed represents RefList L0 (Decoder VLD mode) or Ref Idx Mapping Table L0 (Encoder PAK mode)	1	RefPicList1
Value	Name	Description								
0	RefPicList 0	The list that followed represents RefList L0 (Decoder VLD mode) or Ref Idx Mapping Table L0 (Encoder PAK mode)								
1	RefPicList1	The list that followed represents RefList L1 (Decoder VLD mode) or Ref Idx Mapping Table L1 (Encoder PAK mode)								
2..9	31:0	<p>Reference List Entry This set of fields is always present whenever this command is issued. 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</p> <ul style="list-style-type: none"> • 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) <p>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 :</p> <ul style="list-style-type: none"> • 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 intel implementation) <p>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 the intel specification. If the reference picture is a frame (Bit5 = 1), frame store ID is always an even number.</p>								



MFX_AVC_REF_IDX_STATE

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

MFX_AVC_SLICE_STATE			
Source:	VideoCS		
Length Bias:	2		
Description			
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	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_AVC_SLICE_STATE
		Format:	OpCode
	26:24	Command Opcode	
		Default Value:	1h AVC
		Format:	OpCode
	23:21	SubOpcodeA	
Default Value:		0h MFX_AVC_SLICE_STATE	
Format:		OpCode	
20:16	Command SubOpcodeB		
	Default Value:	3h MFX_AVC_SLICE_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	8h DWORD_COUNT_n	
	Format:	=n	
	Excludes DWords 0,1		
1	31:4	Reserved	
		Format:	MBZ



MFX_AVC_SLICE_STATE											
3:0	<p>Slice Type It is set to the value of the syntax element read from the Slice Header.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0000b</td> <td>P Slice</td> </tr> <tr> <td style="text-align: center;">0001b</td> <td>B Slice</td> </tr> <tr> <td style="text-align: center;">0010b</td> <td>I Slice</td> </tr> <tr> <td style="text-align: center;">0011b-1111b</td> <td>Reserved</td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 10px;">Programming Notes</p> <p>Bits[3:2] must be 0</p>	Value	Name	0000b	P Slice	0001b	B Slice	0010b	I Slice	0011b-1111b	Reserved
	Value	Name									
	0000b	P Slice									
	0001b	B Slice									
	0010b	I Slice									
	0011b-1111b	Reserved									
2	<p>31:30 Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ									
	<p>29:24 Number of Reference Pictures in Inter-prediction List 1</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U6</td> </tr> </table> <p>This field is valid only for encoding a B Slice, for which it is expected to have at least one entry in the reference list L1; otherwise (if Slice Type is not a B Slice), this field must be set to 0. This field can be derived for a B Slice from the Slice Header syntax element NumRefIdxActiveMinus1 as, Num_Ref_Idx_L1 = NumRefIdxActiveMinus1[1] + 1.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0-32</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	0-32					
	Format:	U6									
	Value	Name									
	0-32										
	<p>23:22 Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
<p>21:16 Number of Reference Pictures in Inter-prediction List 0</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U6</td> </tr> </table> <p>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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0-32</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	0-32						
Format:	U6										
Value	Name										
0-32											
<p>15:11 Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ										
10:8	<p>Log 2 Weight Denom Chroma</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U3</td> </tr> </table>	Format:	U3								
	Format:	U3									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0-7</td> <td></td> </tr> </tbody> </table>	Value	Name	0-7								
Value	Name										
0-7											



MFX_AVC_SLICE_STATE		
	7:3	Reserved
		Format: MBZ
	2:0	Log 2 Weight Denom Luma
		Format: U3
		Description
		It is the base 2 logarithm of the denominator for all Luma weighting factors. It is set to the value of the syntax element read from the Slice Header Pred_Weight_Table().
		Value
		Name
		0-7
3	31:30	<p>Weighted Prediction Indicator</p> <p>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.</p> <ul style="list-style-type: none"> If it is a B-Slice, these bits are interpreted as: <ul style="list-style-type: none"> 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) If it is a P Slice, these bits are interpreted as: <ul style="list-style-type: none"> 00b - Disables weighted inter-prediction (Default weighted) 01b - Enables weighted inter-prediction (Explicit weighted) 10b - 11b - Reserved <p style="text-align: center;">Programming Notes</p> <p>Only when in B Slice with Weighted_Pred_Idc = 1 (explicit weighted prediction), will there be a L1 and/or a L0 weight+offset tables being sent to the BSD unit through the Slice_State command.</p> <p>Only when in P Slice with Weighted_Pred_Idc = 1, will there be a L0 weight+offset table being sent to the BSD.</p> <p>If Weighted_Pred_Idc != 1 for B Slice or Weighted_Pred_Idc =0 for P Slice, no Slice_State command should be issued to send these tables. If still being issued, the data is read but ignored.</p> <p>DXVA specifies Weighted_Bipred and Weighted_Pred in frame-level state. However, these two flags are combined and specified in slice level for both P and B slice type.</p>



MFX_AVC_SLICE_STATE		
29	Direct Prediction Type	
	Type of direct prediction used for B Slices. This field is valid only for Slice_Type = B Slice; otherwise, it must be set to 0.	
	Value	Name
	0	Temporal
	1	Spatial
28:27	Disable Deblocking Filter Indicator	
	Value	Name
	00b	FilterInternalEdgesFlag is set equal to 1
	01b	Disable all deblocking operation, no deblocking parameter syntax element is read; filterInternalEdgesFlag is set equal to 0
	10b	Macroblocks in different slices are considered not available; filterInternalEdgesFlag is set equal to 1
11b	Reserved Not defined in AVC	
26	Reserved	
	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	
Programming Notes		
Cabac initialization is also dependent on the field/frame picture type, Slice type, and the current SliceQP value.		
23:22	Reserved	
	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



MFX_AVC_SLICE_STATE											
	<table border="1"> <tr> <td style="text-align: center;">11:8</td> <td>Slice Beta Offset Div2</td> </tr> <tr> <td>Format:</td> <td>S3 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-6, 6] Inclusive</td> </tr> <tr> <td colspan="2">Specifies the offset used in accessing the deblocking filter strength tables.</td> </tr> </table>	11:8	Slice Beta Offset Div2	Format:	S3 2's Complement	Range: [-6, 6] Inclusive		Specifies the offset used in accessing the deblocking filter strength tables.			
	11:8	Slice Beta Offset Div2									
	Format:	S3 2's Complement									
	Range: [-6, 6] Inclusive										
Specifies the offset used in accessing the deblocking filter strength tables.											
<table border="1"> <tr> <td style="text-align: center;">7:4</td> <td>Reserved</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	7:4	Reserved	Format:	MBZ							
7:4	Reserved										
Format:	MBZ										
<table border="1"> <tr> <td style="text-align: center;">3:0</td> <td>Slice Alpha C0 Offset Div2</td> </tr> <tr> <td>Format:</td> <td>S3 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-6, 6] Inclusive</td> </tr> <tr> <td colspan="2">Specifies the offset used in accessing the deblocking filter strength tables.</td> </tr> </table>	3:0	Slice Alpha C0 Offset Div2	Format:	S3 2's Complement	Range: [-6, 6] Inclusive		Specifies the offset used in accessing the deblocking filter strength tables.				
3:0	Slice Alpha C0 Offset Div2										
Format:	S3 2's Complement										
Range: [-6, 6] Inclusive											
Specifies the offset used in accessing the deblocking filter strength tables.											
4	<table border="1"> <tr> <td style="text-align: center;">31:24</td> <td>Slice Vertical Position</td> </tr> <tr> <td colspan="2"> <p>This field specifies the position in y-direction of the first macroblock in the Slice in unit of macroblocks.</p> <p>The fields (Slice_MB_Start_Hor_Pos, Slice_MB_Start_Vert_Pos) are valid in VLD (decoding) mode only. They are ignored by hardware in decoding IT mode and encoding mode (whereas the position is provided by the per-macroblock object command).</p> <p>Derived</p> </td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">Error Handling: Driver needs to check if FirstMbY starts at 0 on the first slice of frame. If not, driver needs to add a phantom slice with FirstMbX and FirstMbY set to 0.</td> </tr> </table>	31:24	Slice Vertical Position	<p>This field specifies the position in y-direction of the first macroblock in the Slice in unit of macroblocks.</p> <p>The fields (Slice_MB_Start_Hor_Pos, Slice_MB_Start_Vert_Pos) are valid in VLD (decoding) mode only. They are ignored by hardware in decoding IT mode and encoding mode (whereas the position is provided by the per-macroblock object command).</p> <p>Derived</p>		Programming Notes		Error Handling: Driver needs to check if FirstMbY starts at 0 on the first slice of frame. If not, driver needs to add a phantom slice with FirstMbX and FirstMbY set to 0.			
31:24	Slice Vertical Position										
<p>This field specifies the position in y-direction of the first macroblock in the Slice in unit of macroblocks.</p> <p>The fields (Slice_MB_Start_Hor_Pos, Slice_MB_Start_Vert_Pos) are valid in VLD (decoding) mode only. They are ignored by hardware in decoding IT mode and encoding mode (whereas the position is provided by the per-macroblock object command).</p> <p>Derived</p>											
Programming Notes											
Error Handling: Driver needs to check if FirstMbY starts at 0 on the first slice of frame. If not, driver needs to add a phantom slice with FirstMbX and FirstMbY set to 0.											
	<table border="1"> <tr> <td style="text-align: center;">23:16</td> <td>Slice Horizontal Position</td> </tr> <tr> <td colspan="2"> <p>This field specifies the position in x-direction of the first macroblock in the Slice in unit of macroblocks.</p> <p>Derived</p> </td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">Error Handling: Driver needs to check if FirstMbY starts at 0 on the first slice of frame. If not, driver needs to add a phantom slice with FirstMbX and FirstMbY set to 0.</td> </tr> </table>	23:16	Slice Horizontal Position	<p>This field specifies the position in x-direction of the first macroblock in the Slice in unit of macroblocks.</p> <p>Derived</p>		Programming Notes		Error Handling: Driver needs to check if FirstMbY starts at 0 on the first slice of frame. If not, driver needs to add a phantom slice with FirstMbX and FirstMbY set to 0.			
23:16	Slice Horizontal Position										
<p>This field specifies the position in x-direction of the first macroblock in the Slice in unit of macroblocks.</p> <p>Derived</p>											
Programming Notes											
Error Handling: Driver needs to check if FirstMbY starts at 0 on the first slice of frame. If not, driver needs to add a phantom slice with FirstMbX and FirstMbY set to 0.											
	<table border="1"> <tr> <td style="text-align: center;">15</td> <td>Reserved</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	15	Reserved	Format:	MBZ						
15	Reserved										
Format:	MBZ										
	<table border="1"> <tr> <td style="text-align: center;">14:0</td> <td>Slice Start Mb Num</td> </tr> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td colspan="2"> <p>The MB number (linear MB address in a picture) at the start of a Slice, it must match with the Slice Horizontal Position (Slice_MB_Start_Hor_Pos) and Vertical Position (Slice_MB_Start_Vert_Pos) in the picture.</p> </td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">In creating the Phantom Slice for error concealment, this field should set to the total number of MB in the current picture + 1.</td> </tr> </table>	14:0	Slice Start Mb Num	Exists If:	//Decoder Only	<p>The MB number (linear MB address in a picture) at the start of a Slice, it must match with the Slice Horizontal Position (Slice_MB_Start_Hor_Pos) and Vertical Position (Slice_MB_Start_Vert_Pos) in the picture.</p>		Programming Notes		In creating the Phantom Slice for error concealment, this field should set to the total number of MB in the current picture + 1.	
14:0	Slice Start Mb Num										
Exists If:	//Decoder Only										
<p>The MB number (linear MB address in a picture) at the start of a Slice, it must match with the Slice Horizontal Position (Slice_MB_Start_Hor_Pos) and Vertical Position (Slice_MB_Start_Vert_Pos) in the picture.</p>											
Programming Notes											
In creating the Phantom Slice for error concealment, this field should set to the total number of MB in the current picture + 1.											



MFX_AVC_SLICE_STATE																
5	31:24	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
	23:16	<p>Next Slice Vertical Position</p> <p>This field specifies the position in y-direction of the first macroblock in the next Slice in unit of macroblocks.</p> <p>This field is primarily used for error concealment. In the case that current slice is the last slice, this field should set to the height of picture (since y-direction is zero-based numbering).</p>														
	15:8	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ															
7:0	<p>Next Slice Horizontal Position</p> <p>This field specifies the position in x-direction of the first macroblock in the next Slice in unit of macroblocks.</p> <p>This field is primarily used for error concealment. In the case that current slice is the last slice, this field should set to 0.</p>															
6 Encoder Only	31	<p>Rate Control Counter Enable</p> <p>To enable the accumulation of bit allocation for rate control</p> <p>This field enables hardware Rate Control logic. The rest of the RC control fields are only valid when this field is set to 1. Otherwise, hardware ignores these fields.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> </tr> <tr> <td>1</td> <td>Enable</td> </tr> </tbody> </table>	Value	Name	0	Disable	1	Enable								
		Value	Name													
		0	Disable													
	1	Enable														
	30	<p>ResetRateControlCounter</p> <p>To reset the bit allocation accumulation counter to 0 to restart the rate control.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Not Reset</td> </tr> <tr> <td>1</td> <td>Reset</td> </tr> </tbody> </table>	Value	Name	0	Not Reset	1	Reset								
		Value	Name													
		0	Not Reset													
	1	Reset														
	<p>RC Triggler Mode</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Always Rate Control</td> <td>Whereas RC becomes active if sum_act > sum_target or sum_act < sum_target</td> </tr> <tr> <td>01b</td> <td>Gentle Rate Control</td> <td>whereas RC becomes active if sum_act > upper_midpt or sum_act < lower_midpt</td> </tr> <tr> <td>10b</td> <td>Loose Rate Control</td> <td>whereas RC becomes active if sum_act > sum_max or sum_act < sum_min</td> </tr> <tr> <td>11b</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	00b	Always Rate Control	Whereas RC becomes active if sum_act > sum_target or sum_act < sum_target	01b	Gentle Rate Control	whereas RC becomes active if sum_act > upper_midpt or sum_act < lower_midpt	10b	Loose Rate Control	whereas RC becomes active if sum_act > sum_max or sum_act < sum_min	11b	Reserved	
	Value	Name	Description													
00b	Always Rate Control	Whereas RC becomes active if sum_act > sum_target or sum_act < sum_target														
01b	Gentle Rate Control	whereas RC becomes active if sum_act > upper_midpt or sum_act < lower_midpt														
10b	Loose Rate Control	whereas RC becomes active if sum_act > sum_max or sum_act < sum_min														
11b	Reserved															
29:28																



MFX_AVC_SLICE_STATE							
27:24	RC Stable Tolerance						
	Format: U4						
	This field specifies the tolerance required to deactivate RC once it has been triggered.						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0-15</td> <td></td> </tr> </tbody> </table>	Value	Name	0-15			
Value	Name						
0-15							
23	RC Panic Enable						
	If this field is set to 1, RC enters panic mode when $sum_act > sum_max$. RC Panic Type field controls what type of panic behavior is invoked.						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> </tr> <tr> <td>1</td> <td>Enable</td> </tr> </tbody> </table>	Value	Name	0	Disable	1	Enable
	Value	Name					
0	Disable						
1	Enable						
22	RC Panic Type						
	This field selects between two RC Panic methods						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>QP Panic</td> </tr> <tr> <td>1</td> <td>CBP Panic</td> </tr> </tbody> </table>	Value	Name	0	QP Panic	1	CBP Panic
	Value	Name					
	0	QP Panic					
	1	CBP Panic					
Programming Notes							
<p>If it is set to 0, in panic mode, the macroblock QP is maxed out, setting to requested QP + QP_max_pos_mod.</p> <p>If it is set to 1, for an intra macroblock, AC CBPs are set to zero (note that DC CBPs are not modified).</p> <p>For inter macroblocks, AC and DC CBPs are forced to zero.</p>							
21	MB Type Direct Conversion Disable						
	Exists If: //B-Slice						
	For all Macroblock type conversions in different slices, refer to Section "Macroblock Type Conversion Rules" in the same volume.						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%; text-align: center;">Value</th> <th style="width: 70%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Enable direct mode conversion</td> </tr> <tr> <td>1</td> <td>Disable direct mode conversion</td> </tr> </tbody> </table>	Value	Name	0	Enable direct mode conversion	1	Disable direct mode conversion
	Value	Name					
	0	Enable direct mode conversion					
1	Disable direct mode conversion						
Programming Notes							
This field is zero for all other slices other than B-Slice.							



MFX_AVC_SLICE_STATE			
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.		
	Value	Name	
	0	Enable skip type conversion	
1	Disable skip type conversion		
Programming Notes			
This field is zero for all other slices other than P_Slice or B-Slice. \			
19	Is Last Slice		
	Description		
	It is used by the zero filling in the Minimum Frame Size test.		
	Set this only for the last slice group		
	Value	Name	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 slice encoded bits.
	1		Header insertion into the output bitstream buffer is present, and is in front of the current slice encoded bits.
	Programming Notes		
This need to be set only for super slice0.			
16	SliceData Insertion Present in Bitstream		
	Value	Name	Description
	0		No Slice Data insertion into the output bitstream buffer
	1		Slice Data insertion into the output bitstream buffer is present.
	Programming Notes		
This bit should be set for all super-slices.			



MFX_AVC_SLICE_STATE											
	15	<p>Tail Insertion Present in bitstream</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>No tail insertion into the output bitstream buffer, after the current slice encoded bits</td> </tr> <tr> <td>1</td> <td></td> <td>Tail insertion into the output bitstream buffer is present, and is after the current slice encoded bits.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This bit should only be set for the last super slice.</p>	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.
	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	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	13	<p>EmulationByteSliceInsertEnable</p> <p>To have PAK outputting SODB or EBSP to the output bitstream buffer</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>outputting RBSP</td> </tr> <tr> <td>1</td> <td></td> <td>outputting EBSP</td> </tr> </tbody> </table>	Value	Name	Description	0		outputting RBSP	1		outputting EBSP
	Value	Name	Description								
	0		outputting RBSP								
	1		outputting EBSP								
12	<p>CabacZeroWordInsertionEnable</p> <p>To pad the end of a SliceLayer RBSP to meet the encoded size requirement.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>No Cabac_Zero_Word Insertion</td> </tr> <tr> <td>1</td> <td></td> <td>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.</td> </tr> </tbody> </table>	Value	Name	Description	0		No Cabac_Zero_Word Insertion	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.	
Value	Name	Description									
0		No Cabac_Zero_Word Insertion									
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	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
7:4	<p>Slice ID [3:0]</p> <p>To identify the output data (coding information record) returned for rate control from PAK to ENC and VPP.</p>										
3:2	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
1:0	<p>Stream ID [1:0]</p> <p>To identify the output data (coding information record) returned for rate control from PAK to ENC and VPP.</p>										
7 Encoder	<p>31:29 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										



MFX_AVC_SLICE_STATE								
Only	28:0	<p>Indirect PAK-BSE Data Start Address (Write)</p> <table border="1"> <tr> <td>Exists If:</td> <td>//AVC Encode Mode</td> </tr> </table> <p>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.</p> <p>It is a byte-aligned address for the AVC bitstream data in both CABAC/CAVLC Modes.</p> <p>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.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0 - 512MB</td> <td></td> </tr> </tbody> </table>	Exists If:	//AVC Encode Mode	Value	Name	0 - 512MB	
		Exists If:	//AVC Encode Mode					
Value	Name							
0 - 512MB								
8 Encoder Only	31:24	<p>Magnitude of QP Max Negative Modifier</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>This field specifies the lower limit of the QP modifier.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0-51</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	0-51	
		Format:	U8					
		Value	Name					
	0-51							
	23:16	<p>Magnitude of QP Max Positive Modifier</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>This field specifies the upper limit of the QP modifier.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0 - 15</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	0 - 15	
		Format:	U8					
		Value	Name					
	0 - 15							
	15:12	<p>Shrink Param - Shrink Resistance</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the additional points added each time decreased correction is invoked.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0 - 15</td> <td></td> </tr> </tbody> </table>	Format:	U4	Value	Name	0 - 15	
		Format:	U4					
		Value	Name					
	0 - 15							
11:8	<p>Shrink Param - Shrink Init</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the initial points required to trip decreased control.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0 - 15</td> <td></td> </tr> </tbody> </table>	Format:	U4	Value	Name	0 - 15		
	Format:	U4						
	Value	Name						
0 - 15								
7:4	<p>Grow Param - Grow Resistance</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the additional points added each time increased correction is invoked.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0 - 15</td> <td></td> </tr> </tbody> </table>	Format:	U4	Value	Name	0 - 15		
	Format:	U4						
	Value	Name						
0 - 15								



MFX_AVC_SLICE_STATE																			
	3:0	Grow Param - Grow Init Format: U4 This field specifies the initial points required to trip increased control.																	
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0 - 15</td> <td></td> </tr> </tbody> </table>	Value	Name	0 - 15														
Value	Name																		
0 - 15																			
9 Encoder Only	31	RoundInterEnable Format: Enable When this bit is not set, RoundInter defaults to 2 to match SNB.																	
		RoundInter Format: U3 Rounding precision for Inter quantized coefficients <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%; text-align: center;">Value</th> <th style="width: 70%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr><td style="text-align: center;">000b</td><td style="text-align: center;">+1/16 [Default]</td></tr> <tr><td style="text-align: center;">001b</td><td style="text-align: center;">+2/16</td></tr> <tr><td style="text-align: center;">010b</td><td style="text-align: center;">+3/16</td></tr> <tr><td style="text-align: center;">011b</td><td style="text-align: center;">+4/16</td></tr> <tr><td style="text-align: center;">100b</td><td style="text-align: center;">+5/16</td></tr> <tr><td style="text-align: center;">101b</td><td style="text-align: center;">+6/16</td></tr> <tr><td style="text-align: center;">110b</td><td style="text-align: center;">+7/16</td></tr> <tr><td style="text-align: center;">111b</td><td style="text-align: center;">+8/16</td></tr> </tbody> </table>	Value	Name	000b	+1/16 [Default]	001b	+2/16	010b	+3/16	011b	+4/16	100b	+5/16	101b	+6/16	110b	+7/16	111b
	Value	Name																	
	000b	+1/16 [Default]																	
	001b	+2/16																	
	010b	+3/16																	
	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 to 4 to match SNB.																		
	RoundIntra Format: U3 Rounding precision for Intra quantized coefficients <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%; text-align: center;">Value</th> <th style="width: 70%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr><td style="text-align: center;">000b</td><td style="text-align: center;">+1/16 [Default]</td></tr> <tr><td style="text-align: center;">001b</td><td style="text-align: center;">+2/16</td></tr> <tr><td style="text-align: center;">010b</td><td style="text-align: center;">+3/16</td></tr> <tr><td style="text-align: center;">011b</td><td style="text-align: center;">+4/16</td></tr> <tr><td style="text-align: center;">100b</td><td style="text-align: center;">+5/16</td></tr> <tr><td style="text-align: center;">101b</td><td style="text-align: center;">+6/16</td></tr> <tr><td style="text-align: center;">110b</td><td style="text-align: center;">+7/16</td></tr> <tr><td style="text-align: center;">111b</td><td style="text-align: center;">+8/16</td></tr> </tbody> </table>	Value	Name	000b	+1/16 [Default]	001b	+2/16	010b	+3/16	011b	+4/16	100b	+5/16	101b	+6/16	110b	+7/16	111b	+8/16
Value	Name																		
000b	+1/16 [Default]																		
001b	+2/16																		
010b	+3/16																		
011b	+4/16																		
100b	+5/16																		
101b	+6/16																		
110b	+7/16																		
111b	+8/16																		



MFX_AVC_SLICE_STATE										
	23:20	<p>Correct 6</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> <tr> <td colspan="2">This field specifies the points used in the lowermost RC region when $\text{sum_act} \leq \text{sum_min}$.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>0 - 15</td> <td></td> </tr> </table>	Format:	U4	This field specifies the points used in the lowermost RC region when $\text{sum_act} \leq \text{sum_min}$.		Value	Name	0 - 15	
	Format:	U4								
	This field specifies the points used in the lowermost RC region when $\text{sum_act} \leq \text{sum_min}$.									
	Value	Name								
	0 - 15									
19:16	<p>Correct 5</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> <tr> <td colspan="2">This field specifies the points used in the fifth RC region when $\text{sum_act} > \text{sum_min}$ but $\leq \text{lower_midpt}$.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>0 - 15</td> <td></td> </tr> </table>	Format:	U4	This field specifies the points used in the fifth RC region when $\text{sum_act} > \text{sum_min}$ but $\leq \text{lower_midpt}$.		Value	Name	0 - 15		
Format:	U4									
This field specifies the points used in the fifth RC region when $\text{sum_act} > \text{sum_min}$ but $\leq \text{lower_midpt}$.										
Value	Name									
0 - 15										
15:12	<p>Correct 4</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> <tr> <td colspan="2">This field specifies the points used in the fourth RC region when $\text{sum_act} > \text{lower_midpt}$ but $\leq \text{sum_target}$.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>0 - 15</td> <td></td> </tr> </table>	Format:	U4	This field specifies the points used in the fourth RC region when $\text{sum_act} > \text{lower_midpt}$ but $\leq \text{sum_target}$.		Value	Name	0 - 15		
Format:	U4									
This field specifies the points used in the fourth RC region when $\text{sum_act} > \text{lower_midpt}$ but $\leq \text{sum_target}$.										
Value	Name									
0 - 15										
11:8	<p>Correct 3</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> <tr> <td colspan="2">This field specifies the points used in the third RC region when $\text{sum_act} > \text{sum_target}$ but $\leq \text{upper_midpt}$.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>0 - 15</td> <td></td> </tr> </table>	Format:	U4	This field specifies the points used in the third RC region when $\text{sum_act} > \text{sum_target}$ but $\leq \text{upper_midpt}$.		Value	Name	0 - 15		
Format:	U4									
This field specifies the points used in the third RC region when $\text{sum_act} > \text{sum_target}$ but $\leq \text{upper_midpt}$.										
Value	Name									
0 - 15										
7:4	<p>Correct 2</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> <tr> <td colspan="2">This field specifies the points used in the second RC region when $\text{sum_act} > \text{upper_midpt}$ but $\leq \text{sum_max}$.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>0 - 15</td> <td></td> </tr> </table>	Format:	U4	This field specifies the points used in the second RC region when $\text{sum_act} > \text{upper_midpt}$ but $\leq \text{sum_max}$.		Value	Name	0 - 15		
Format:	U4									
This field specifies the points used in the second RC region when $\text{sum_act} > \text{upper_midpt}$ but $\leq \text{sum_max}$.										
Value	Name									
0 - 15										
3:0	<p>Correct 1</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> <tr> <td colspan="2">This field specifies the points used in the topmost RC region when $\text{sum_act} > \text{sum_max}$.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>0 - 15</td> <td></td> </tr> </table>	Format:	U4	This field specifies the points used in the topmost RC region when $\text{sum_act} > \text{sum_max}$.		Value	Name	0 - 15		
Format:	U4									
This field specifies the points used in the topmost RC region when $\text{sum_act} > \text{sum_max}$.										
Value	Name									
0 - 15										
10 Encoder Only	31:28	ClampValues - CV7								
	27:24	CV6								
	23:20	CV5								



MFX_AVC_SLICE_STATE

19:16	CV4																																																																																																																																																		
15:12	CV3																																																																																																																																																		
11:8	CV2																																																																																																																																																		
7:4	CV1																																																																																																																																																		
3:0	CV0 - Clamp Value 0																																																																																																																																																		
	Format:	U4																																																																																																																																																	
	<p>If the magnitude of coefficients at locations assigned with CV0 (mapping shown below) exceeds 2^{CV0-1}, they are replaced with 2^{CV0-1}. For coefficients at locations marked as 'none', no clamping is performed. The following mappings are only applied to luma and chroma blocks/subblocks containing AC coefficients (blocks/subblocks with only DC coeffs will not be clamped).</p> <p>For 4x4 frame block, each coefficient is mapped to one of the eight CV values as following:</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>none</td><td>CV7</td><td>CV5</td><td>CV4</td></tr> <tr><td>CV7</td><td>CV6</td><td>CV4</td><td>CV3</td></tr> <tr><td>CV5</td><td>CV4</td><td>CV2</td><td>CV1</td></tr> <tr><td>CV4</td><td>CV3</td><td>CV1</td><td>CV0</td></tr> </table> <p>For 8x8 frame block, each coefficient is mapped to one of the eight CV values as following:</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>none</td><td>none</td><td>CV7</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td></tr> <tr><td>none</td><td>CV7</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td></tr> <tr><td>CV7</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td><td>CV2</td></tr> <tr><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td><td>CV2</td><td>CV1</td></tr> <tr><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td><td>CV2</td><td>CV1</td><td>CV1</td></tr> <tr><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td><td>CV2</td><td>CV1</td><td>CV1</td><td>CV0</td></tr> <tr><td>CV3</td><td>CV3</td><td>CV2</td><td>CV2</td><td>CV1</td><td>CV1</td><td>CV0</td><td>CV0</td></tr> <tr><td>CV3</td><td>CV2</td><td>CV2</td><td>CV1</td><td>CV1</td><td>CV0</td><td>CV0</td><td>CV0</td></tr> </table> <p>For 4x4 field block, each coefficient is mapped to one of the eight CV values as following:</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>none</td><td>CV6</td><td>CV3</td><td>CV1</td></tr> <tr><td>CV7</td><td>CV6</td><td>CV3</td><td>CV1</td></tr> <tr><td>CV5</td><td>CV4</td><td>CV2</td><td>CV0</td></tr> <tr><td>CV5</td><td>CV4</td><td>CV2</td><td>CV0</td></tr> </table> <p>For 8x8 field block, each coefficient is mapped to one of the eight CV values as following:</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>none</td><td>none</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV2</td><td>CV1</td></tr> <tr><td>none</td><td>CV7</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV2</td><td>CV1</td></tr> <tr><td>CV7</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV3</td><td>CV2</td><td>CV1</td></tr> <tr><td>CV7</td><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV2</td><td>CV2</td><td>CV1</td></tr> <tr><td>CV6</td><td>CV5</td><td>CV4</td><td>CV4</td><td>CV3</td><td>CV2</td><td>CV1</td><td>CV0</td></tr> <tr><td>CV6</td><td>CV5</td><td>CV4</td><td>CV3</td><td>CV2</td><td>CV2</td><td>CV1</td><td>CV0</td></tr> </table>			none	CV7	CV5	CV4	CV7	CV6	CV4	CV3	CV5	CV4	CV2	CV1	CV4	CV3	CV1	CV0	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	none	CV6	CV3	CV1	CV7	CV6	CV3	CV1	CV5	CV4	CV2	CV0	CV5	CV4	CV2	CV0	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
none	CV7	CV5	CV4																																																																																																																																																
CV7	CV6	CV4	CV3																																																																																																																																																
CV5	CV4	CV2	CV1																																																																																																																																																
CV4	CV3	CV1	CV0																																																																																																																																																
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																																																																																																																																												
none	CV6	CV3	CV1																																																																																																																																																
CV7	CV6	CV3	CV1																																																																																																																																																
CV5	CV4	CV2	CV0																																																																																																																																																
CV5	CV4	CV2	CV0																																																																																																																																																
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																																																																																																																																												



MFX_AVC_SLICE_STATE										
		CV5	CV5	CV4	CV3	CV2	CV1	CV1	CV0	
		CV5	CV5	CV4	CV3	CV2	CV1	CV1	CV0	
		Value					Name			
		0 - 15								



MFX_AVC_WEIGHTOFFSET_STATE

MFX_AVC_WEIGHTOFFSET_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>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)</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_AVC_WEIGHTOFFSET_STATE
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	1h AVC_COMMON
		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 Excludes DWord (0,1)	
	Format:	=n Total Length - 2	



MFX_AVC_WEIGHTOFFSET_STATE											
1	31:1	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
0	<p>Weight and Offset Select</p> <p>It must be set in consistent with the WeightedPredFlag and WeightedBiPredIdc in the lmg_State command.</p> <p>This parameter is specified for Intel interface only, not present in the DXVA.</p> <p>For implicit even though only one entry may be used, still loading the whole 32-entry table.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Weight and Offset L0 table</td> <td>The list that followed is associated with the weight and offset for RefPicList L0</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Weight and Offset L1 table</td> <td>The list that followed is associated with the weight and offset for RefPicList L1</td> </tr> </tbody> </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
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									
2..97	31:0	<p>WeightOffset</p> <p>WeightOffset[L=L0=0 or L1=1][i=0 to 31][Y=0/Cb=1/Cr=2][weight=0/offset=1] WeightOffset[L][i=0][Y=0][Weight=0], WeightOffset[L][i=0][Y=0][Offset=1] WeightOffset[L][i=0][Cb=1][Weight=0], WeightOffset[L][i=0][Cb=1][Offset=1] WeightOffset[L][i=0][Cr=2][Weight=0], WeightOffset[L][i=0][Cr=2][Offset=1]: WeightOffset[L][i=31][Y=0][Weight=0], WeightOffset[L][i=31][Y=0][Offset=1] WeightOffset[L][i=31][Cb=1][Weight=0], WeightOffset[L][i=31][Cb=1][Offset=1] WeightOffset[L][i=31][Cr=2][Weight=0], WeightOffset[L][i=31][Cr=2][Offset=1]</p> <p>Format for explicit: Both Weight and Offset are S15 in two's compliment, with a valid range from -128 to 128 Format for implicit: S15</p> <p>This set of fields is always present whenever this command is issued. The full table, one entry for each reference picture, is always specified. Any reference list L0/L1[i] that does not exist, the corresponding weight and offset are set to 0.</p> <p>Weight and Offset are 2 byte each. Apair of Weight and Offset forms a dword, with Weight in the LOWER word and Offset in the HIGHER word.</p> <p>WeightOffset[L0=0][i=0 to 31][Y=0] (i.e. luma_weight_l0[i]) are specified for the weighting and offset factors applied to the luma prediction value for list 0 prediction using RefPicList0[i] (one-to-one correspondence in i). When luma_weight_l0_flag (Slice Header syntax element) is equal to 1, the value of luma_weight_l0[i] shall be in the range of -128 to 127. When luma_weight_l0_flag is equal to 0, luma_weight_l0[i] shall be inferred to be equal to 2luma_log2_weight_denom for RefPicList0[i]. luma_log2_weight_denom is a Slice Header syntax element.</p> <p>WeightOffset[L0=0][i=0 to 31][Cb=1] (i.e. chromaCb_weight_l0[i]) are specified for the weighting and offset factors applied to the chroma Cb prediction values for list 0 prediction using RefPicList0[i] (one-to-one correspondence in i). When chroma_weight_l0_flag (Slice Header syntax element) is equal to 1, the value of chromaCb_weight_l0[i] shall be in the range of -128 to 127. When chroma_weight_l0_flag is equal to 0, chromaCb_weight_l0[i] shall be inferred to be equal to 2chroma_log2_weight_denom for RefPicList0[i]. chroma_log2_weight_denom is a Slice Header syntax element.</p> <p>WeightOffset[L0=0][i=0 to 31][Cr=2] (i.e. chromaCr_weight_l0[i]) are specified for the</p>									



MFX_AVC_WEIGHTOFFSET_STATE

	<p>weighting and offset factors applied to the chroma Cr prediction values for list 0 prediction using RefPicList0[i] (one-to-one correspondence in i). When chroma_weight_l0_flag (Slice Header syntax element) is equal to 1, the value of chromaCr_weight_l0[i] shall be in the range of -128 to 127. When chroma_weight_l0_flag is equal to 0, chromaCr_weight_l0[i] shall be inferred to be equal to $2^{\text{chroma_log2_weight_denom}}$ for RefPicList0[i].</p>
--	---



MFX_BSP_BUF_BASE_ADDR_STATE

MFX_BSP_BUF_BASE_ADDR_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>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)</p> <p>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).</p> <p>In addition, in Encoder, the row stores for CABAC encoding and MB Parameters Construction (MPC) are combined into one single row store.</p> <p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Pipeline
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	0h MFX_COMMON_STATE
		Format:	OpCode
	23:21	SubOpcode A	
		Default Value:	0h
		Format:	OpCode
	20:16	SubOpcode B	
		Default Value:	4h
Format:		OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	8h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	



MFX_BSP_BUF_BASE_ADDR_STATE									
1	31:6	<p>BSD/MPC Row Store Scratch Buffer Base Address - Read/Write</p> <p>This field provides the base address of the scratch buffer used by BSD (decoder) and MPC (encoder) unit to store MB information of the previous row for coding each macroblock in the current row. It is a private buffer used by the BSD (decoder) and MPC (encoder) hardware only. Its content is not accessible by software. This Row Store buffer must be 64-byte cacheline aligned. Hardware uses the horizontal address of the current macroblock to address this Row Store.</p> <p>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.</p> <p style="text-align: center;">Programming Notes</p> <p>This is one of the four RowStore Scratch Buffers which can programmed to use the internal Media Storage (total size 640 CacheLine). When Deblocking Filter Row Store Scratch Buffer Cache Select is programmed to "1", this will be stored inside MFX Media Internal Storage. Driver then needs to program this Base Address between 0 to 639, indicating starting cacheline address location for this buffer. Driver needs to make sure the whole buffer fits into Media Internal Storage.</p> <p><i>(Notes: 1 cachelines per MB for non-mbaff; 2 cachelines per MB pair for mbaff, and the buffer needs to have enough space for 1 MB (pair) row).</i></p>							
	5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ								
2	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table> <p>Reserved for 64-bit address extension.</p>	Format:	MBZ					
	Format:	MBZ							
15:0	<p>BSD/MPC Row Store Scratch Buffer Base Address - Read/Write [47:32]</p> <p style="text-align: center;">Description</p> <p>This field is for the upper range of BSD/MPC Row Store Scratch Buffer Base Address.</p> <p>This field is used for 48-bit addressing.</p>								
3	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
14:13	<p>BSD/MPC Row Store Scratch Buffer - Tiled Resource Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource
Format:	U2								
Value	Name	Description							
0h	TRMODE_NONE	No tiled resource							



MFx_BSP_BUF_BASE_ADDR_STATE		
	1h	TRMODE_TILEYF 4KB tiled resources
	2h	TRMODE_TILEYS 64KB tiled resources
	3h	Reserved
12	BSD/MPC Row Store Scratch Buffer Cache Select This field controls if Intra Row Store is going to store inside Media Internal Storage or to LLC.	
	Value	Name Description
	0	Buffer going to LLC
	1	Buffer going to Internal Media Storage
11	Reserved Format: _____ MBZ	
10:9	Reserved Format: _____ MBZ	
8:7	BSD/MPC Row Store Scratch Buffer - Arbitration Priority Control Format: _____ U2 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
6:1	BSD/MPC Row Store Scratch Buffer - Index to Memory Object Control State (MOCS) Tables Format: _____ U6 The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
0	Reserved	
4	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



MFX_BSP_BUF_BASE_ADDR_STATE																			
		<p>alignment should be followed. This field is only valid for AVC decoder mode</p> <p>This is one of the four RowStore Scratch Buffers which can be programmed to use the internal Media Storage (total size 640 CacheLine). When Deblocking Filter Row Store Scratch Buffer Cache Select is programmed to "1", this will be cache inside MFX Media Internal Storage. Driver then needs to program this Base Address between 0 to 639, indicating starting cachelines address location for this buffer. Driver needs to make sure the whole buffer fits into Media Internal Storage</p> <p><i>(Notes: 1 cachelines per MB for non-mbaff; 2 cachelines per MB pair for mbaff, and the buffer needs to have enough space for 1 MB (pair) row).</i></p>																	
	5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																		
5	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Reserved for 64-bit address extension.</p>	Format:	MBZ															
	Format:	MBZ																	
15:0	<p>MPR Row Store Scratch Buffer Base Address - Read/Write [47:32]</p> <p>This field is for the upper range of MPR Row Store Scratch Buffer Base Address. This field is used for 48-bit addressing.</p>																		
6	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ															
	Format:	MBZ																	
	14:13	<p>MPR Row Store Scratch Buffer - Tiled Resource Mode</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Format:	U2																	
	Value	Name	Description																
0h	TRMODE_NONE	No tiled resource																	
1h	TRMODE_TILEYF	4KB tiled resources																	
2h	TRMODE_TILEYS	64KB tiled resources																	
3h	Reserved																		
12	<p>MPR Row Store Scratch Buffer Cache Select</p> <p>This field controls if Intra Row Store is going to store inside Media Internal Storage or to LLC.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Buffer going to LLC</td> </tr> <tr> <td>1</td> <td></td> <td>Buffer going to Internal Media Storage</td> </tr> </tbody> </table>	Value	Name	Description	0		Buffer going to LLC	1		Buffer going to Internal Media Storage									
Value	Name	Description																	
0		Buffer going to LLC																	
1		Buffer going to Internal Media Storage																	
11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																		
10:9	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																		



MFX_BSP_BUF_BASE_ADDR_STATE														
	8:7	<p>MPR Row Store Scratch Buffer - Arbitration Priority Control</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Format:	U2	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
	Format:	U2												
Value	Name													
00b	Highest priority													
01b	Second highest priority													
10b	Third highest priority													
11b	Lowest priority													
	6:1	<p>MPR Row Store Scratch Buffer - Index to Memory Object Control State (MOCS) Tables</p> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.</p> <p>The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>	Format:	U6										
Format:	U6													
	0	Reserved												
7	31:6	<p>Bitplane Read Buffer Base Address</p> <p>It must be cacheline aligned (i.e. 64 bytes address boundary), so lower bit 0 to 5 are used for controlling information.(In Cantiga, this field must be dword aligned.)Bitplane buffer is a linear buffer. In VC1 Long format, it is written by an application. In VC1 Short Format, it is written and read by H/W only.For VC1 intel Long Format : it is a read-only bufferFor VC1 DXVA2 Short Format : it is a write and a read bufferThis field is only valid for VC1 decoder mode.</p>												
	5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ													
8	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Reserved for 64-bit address extension.</p>	Format:	MBZ										
	Format:	MBZ												
15:0	<p>Bitplane Read Buffer Base Address - Read/Write [47:32]</p> <p>This field is for the upper range of Bitplane Read Buffer Base Address.</p> <p>This field is used for 48-bit addressing.</p>													
9	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
	Format:	MBZ												
14:13	<p>Bitplane Read Buffer - Tiled Resource Mode</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>For Media Surfaces:</p> <p>This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource					
Format:	U2													
Value	Name	Description												
0h	TRMODE_NONE	No tiled resource												



MFX_BSP_BUF_BASE_ADDR_STATE		
	1h	TRMODE_TILEYF 4KB tiled resources
	2h	TRMODE_TILEYS 64KB tiled resources
	3h	Reserved
12:11	Reserved	
	Format:	MBZ
10:9	Reserved	
	Format:	MBZ
8:7	Bitplane Read Buffer - Arbitration Priority Control	
	Format:	U2
	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
6:1	Bitplane Read Buffer - Index to Memory Object Control State (MOCS) Tables	
	Format:	U6
	The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.	
	The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
0	Reserved	



MFX_DBK_OBJECT

MFX_DBK_OBJECT			
Source:		VideoCS	
Length Bias:		2	
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_DBK_OBJECT
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	0h Common
		Format:	OpCode
	23:21	SubOpcode A	
Default Value:		0h	
Format:		OpCode	
20:16	SubOpcode B		
	Default Value:	9h	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	0Bh Excludes DWord (0,1)	
	Format:	=n	
	Note: Regardless of the mode, inline data must be present in this command		
1	31:6	Pre Deblocking Source Address	
		Format:	GraphicsAddress[31:6]
	Specifies the 4K byte aligned frame buffer address for outputting the non-filtered reconstructed YUV picture (i.e. output of final adder in each codec standard, and prior to the deblocking filter unit).		
5:0	Reserved		
	Format:	MBZ	
2	31:16	Reserved	
		Format:	MBZ



MFX_DBK_OBJECT																
	<p>15:0 Pre Deblocking Source Address High</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Pre-Deblocking Source Address. This field is used for 48-bit addressing.</p>	Format:	GraphicsAddress[47:32]													
Format:	GraphicsAddress[47:32]															
3	<p>31:15 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ														
	<p>14:13 Pre Deblocking Source - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description													
	0h	TRMODE_NONE	No tiled resource													
1h	TRMODE_TILEYF	4KB tiled resources														
2h	TRMODE_TILEYS	64KB tiled resources														
3h	Reserved															
<p>12:11 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ															
<p>10 Pre Deblocking Source - Memory Compression Mode Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter, Media Memory Compression section for more details.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	1	Vertical Compression Mode												
Value	Name															
1	Vertical Compression Mode															
<p>9 Pre Deblocking Source - Memory Compression Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Memory compression will be attempted for this surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Compression Disable</td> </tr> <tr> <td>1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	0	Compression Disable	1	Compression Enable								
Format:	Enable															
Value	Name															
0	Compression Disable															
1	Compression Enable															
<p>8:7 Pre Deblocking Source - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority						
Value	Name															
00b	Highest priority															
01b	Second highest priority															
10b	Third highest priority															
11b	Lowest priority															



MFX_DBK_OBJECT																	
	6:1	<p>Pre Deblocking Source - Index to Memory Object Control State (MOCS) Tables</p> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>															
	0	Reserved															
4	31:6	<p>Deblocking Control Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Specifies the 4K byte aligned frame buffer address as input MB-level deblocking parameters to control the way hardware deblock the each micro-block. One 512-bit cacheline is allocated for each Macroblock in raster scan order.</p>	Format:	GraphicsAddress[31:6]													
	Format:	GraphicsAddress[31:6]															
5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
5	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
15:0	<p>Deblocking Control Address High</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Deblocking Control Address (DeblockCntrlAddr). This field is used for 48-bit addressing.</p>	Format:	GraphicsAddress[47:32]														
Format:	GraphicsAddress[47:32]																
6	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	<p>Deblocking Control - Tiled Resource Mode For Media Surfaces:</p> <p>This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
0h	TRMODE_NONE	No tiled resource															
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
12:11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
10	<p>Deblocking Control - Memory Compression Mode</p> <p>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter, Media Memory Compression section for more details.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 85%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td>1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode										
Value	Name																
0	Horizontal Compression Mode																
1	Vertical Compression Mode																



MFX_DBK_OBJECT											
	9	<p>Deblocking Control - Memory Compression Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Memory compression will be attempted for this surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Compression Disable</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	0	Compression Disable			
	Format:	Enable									
Value	Name										
0	Compression Disable										
8:7	<p>Deblocking Control - Arbitration Priority Control</p> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
Value	Name										
00b	Highest priority										
01b	Second highest priority										
10b	Third highest priority										
11b	Lowest priority										
	6:1	<p>Deblocking Source - Index to Memory Object Control State (MOCS) Tables</p> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>									
	0	Reserved									
7	31:6	<p>Deblocking Destination Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Specifies the 4K byte aligned frame buffer address for outputting the post-loop filtered reconstructed YUV picture (i.e. output of the deblocking filter unit)</p>	Format:	GraphicsAddress[31:6]							
	Format:	GraphicsAddress[31:6]									
5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
8	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
15:0	<p>Deblocking Destination Address High</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Deblocking Destination Address. This field is used for 48-bit addressing.</p>	Format:	GraphicsAddress[47:32]								
Format:	GraphicsAddress[47:32]										
9	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
14:13	<p>Deblocking Destination - Tiled Resource Mode</p> <p>For Media Surfaces:</p> <p>This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	
Value	Name	Description									
0h	TRMODE_NONE	No tiled resource									
1h	TRMODE_TILEYF	4KB tiled resources									



MFX_DBK_OBJECT			
	2h	TRMODE_TILEYS	64KB tiled resources
	3h	Reserved	
12:11	Reserved		
	Format:	MBZ	
10	Deblocking Destination - Memory Compression Mode Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter, Media Memory Compression section for more details.		
	Value	Name	
	0	Horizontal Compression Mode	
9	Deblocking Destination - Memory Compression Enable Format: Enable Memory compression will be attempted for this surface.		
	Value	Name	
	0	Compression Disable	
8:7	Deblocking Destination - 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 priority	
	11b	Lowest priority	
6:1	Deblocking Destination - Index to Memory Object Control State (MOCS) Tables The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.		
0	Reserved		
10	31:6	Deblock Row Store Address Format: GraphicsAddress[31:6] This field provides the base address 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 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 Filter Row Store.	
5:0	Reserved		
	Format:	MBZ	



MFX_DBK_OBJECT																	
11	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Reserved for 64-bit address extension.</p>	Format:	MBZ													
	Format:	MBZ															
15:0	<p>Deblock Row Store Address High</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Deblock Row Store Address (DeblockRowStoreAddr). This field is used for 48-bit addressing.</p>	Format:	GraphicsAddress[47:32]														
Format:	GraphicsAddress[47:32]																
12	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	<p>Deblock Row Store - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
	0h	TRMODE_NONE	No tiled resource														
	1h	TRMODE_TILEYF	4KB tiled resources														
	2h	TRMODE_TILEYS	64KB tiled resources														
3h	Reserved																
12	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
10	<p>Deblock Row Store - Memory Compression Mode Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter, Media Memory Compression section for more details.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Horizontal Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode												
Value	Name																
0	Horizontal Compression Mode																
9	<p>Deblock Row Store - Memory Compression Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Memory compression will be attempted for this surface.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Compression Disable</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	0	Compression Disable										
Format:	Enable																
Value	Name																
0	Compression Disable																



MFX_DBK_OBJECT											
8:7	<p>Deblock Row Store - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
Value	Name										
00b	Highest priority										
01b	Second highest priority										
10b	Third highest priority										
11b	Lowest priority										
6:1	<p>CoeffProbability StreamIn Address - Index to Memory Object Control State (MOCS) Tables The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>										
0	Reserved										



MFX_FQM_STATE

MFX_FQM_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_MULTI_DW
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	0h MFX_COMMON_STATE
Format:		OpCode	
23:21	SubOpcode A		
	Default Value:	0h	
	Format:	OpCode	
20:16	SubOpcode B		
	Default Value:	8h	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	20h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:2	Reserved	
		Format:	MBZ



MFX_FQM_STATE														
	1:0	<p>AVC</p> <table border="1"> <tr> <td>Exists If:</td> <td>//AVC- Decoder Only</td> </tr> </table> <p>For AVC QM Type: This field specifies which Quantizer Matrix is loaded.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>AVC_4x4_Intra_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)</td> </tr> <tr> <td>1</td> <td>AVC_4x4_Inter_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)</td> </tr> <tr> <td>2</td> <td>AVC_8x8_Intra_MATRIX</td> </tr> <tr> <td>3</td> <td>AVC_8x8_Inter_MATRIX</td> </tr> </tbody> </table>	Exists If:	//AVC- Decoder Only	Value	Name	0	AVC_4x4_Intra_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)	1	AVC_4x4_Inter_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)	2	AVC_8x8_Intra_MATRIX	3	AVC_8x8_Inter_MATRIX
	Exists If:	//AVC- Decoder Only												
	Value	Name												
	0	AVC_4x4_Intra_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)												
	1	AVC_4x4_Inter_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)												
	2	AVC_8x8_Intra_MATRIX												
	3	AVC_8x8_Inter_MATRIX												
	1:0	<p>MPEG2</p> <table border="1"> <tr> <td>Exists If:</td> <td>//MPEG2- Decoder Only</td> </tr> </table> <p>For MPEG2 QM Type: This field specifies which Quantizer Matrix is loaded.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>MPEG_INTRA_QUANTIZER_MATRIX</td> </tr> <tr> <td>1</td> <td>MPEG_NON_INTRA_QUANTIZER_MATRIX</td> </tr> <tr> <td>2-3</td> <td>Reserved</td> </tr> </tbody> </table>	Exists If:	//MPEG2- Decoder Only	Value	Name	0	MPEG_INTRA_QUANTIZER_MATRIX	1	MPEG_NON_INTRA_QUANTIZER_MATRIX	2-3	Reserved		
	Exists If:	//MPEG2- Decoder Only												
	Value	Name												
	0	MPEG_INTRA_QUANTIZER_MATRIX												
	1	MPEG_NON_INTRA_QUANTIZER_MATRIX												
2-3	Reserved													
1:0	<p>JPEG</p> <table border="1"> <tr> <td>Exists If:</td> <td>//JPEG- Encoder Only</td> </tr> </table> <p>For JPEG QM Type: This field specifies which Quantizer Matrix is loaded.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>JPEG_Luma_Y_QUANTIZER_MATRIX (or R)</td> </tr> <tr> <td>1</td> <td>JPEG_Chroma_Cb_QUANTIZER_MATRIX (or G)</td> </tr> <tr> <td>2</td> <td>JPEG_Chroma_Cr_QUANTIZER_MATRIX (or B)</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>For JPEG encoder, each quantization element presents 16-bit 1/QM[i][j]. In RGB encoding, because the order input image components can be RGB, GBR, BGR, YUV, the value 0 is used for the first image component, the value 1 is used for the second image component, and the value 2 is used for the third image component.</p>	Exists If:	//JPEG- Encoder Only	Value	Name	0	JPEG_Luma_Y_QUANTIZER_MATRIX (or R)	1	JPEG_Chroma_Cb_QUANTIZER_MATRIX (or G)	2	JPEG_Chroma_Cr_QUANTIZER_MATRIX (or B)			
Exists If:	//JPEG- Encoder Only													
Value	Name													
0	JPEG_Luma_Y_QUANTIZER_MATRIX (or R)													
1	JPEG_Chroma_Cb_QUANTIZER_MATRIX (or G)													
2	JPEG_Chroma_Cr_QUANTIZER_MATRIX (or B)													
2..33	31:0	<p>Forward Quantizer Matrix</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>The format of a Quantizer Matrix is an 8x8 matrix in raster order. Each element is an unsigned byte.</p>	Format:	U32										
Format:	U32													



MFX_IND_OBJ_BASE_ADDR_STATE

MFX_IND_OBJ_BASE_ADDR_STATE		
Source:	VideoCS	
Length Bias:	2	
<p>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.</p>		
<p>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.</p>		
<p>While the use of base address is unconditional, the indirection can be effectively disabled by setting the base address to zero.</p> <p>For decoder, there are:</p> <ul style="list-style-type: none"> • 1 read-only per-slice indirect object in the BSD_OBJECT Command, and • 2 read-only per-MB indirect objects in the IT_OBJECT Command. <p>For 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).</p> <p>For encoder, there are:</p> <ul style="list-style-type: none"> • 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 Command <p>For 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 requestor. NotationDefinitionPhysicalAddress[n:m] Corresponding bits of a physical graphics memory byte address (not mapped by a GTT) GraphicsAddress[n:m] Corresponding bits of an absolute, virtual graphics memory byte address (mapped by a GTT).</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
		Format: OpCode



MFX_IND_OBJ_BASE_ADDR_STATE							
	28:27	Pipeline <table border="1"> <tr> <td>Default Value:</td> <td>2h MFX_IND_OBJ_BASE_ADDR_STATE</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	2h MFX_IND_OBJ_BASE_ADDR_STATE	Format:	OpCode	
	Default Value:	2h MFX_IND_OBJ_BASE_ADDR_STATE					
	Format:	OpCode					
	26:24	Common Opcode <table border="1"> <tr> <td>Default Value:</td> <td>0h MFX_IND_OBJ_BASE_ADDR_STATE</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	0h MFX_IND_OBJ_BASE_ADDR_STATE	Format:	OpCode	
	Default Value:	0h MFX_IND_OBJ_BASE_ADDR_STATE					
	Format:	OpCode					
	23:21	Sub OpcodeA <table border="1"> <tr> <td>Default Value:</td> <td>0h MFX_IND_OBJ_BASE_ADDR_STATE</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	0h MFX_IND_OBJ_BASE_ADDR_STATE	Format:	OpCode	
	Default Value:	0h MFX_IND_OBJ_BASE_ADDR_STATE					
	Format:	OpCode					
	20:16	SubOpcodeB <table border="1"> <tr> <td>Default Value:</td> <td>3h MFX_IND_OBJ_BASE_ADDR_STATE</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	3h MFX_IND_OBJ_BASE_ADDR_STATE	Format:	OpCode	
Default Value:	3h MFX_IND_OBJ_BASE_ADDR_STATE						
Format:	OpCode						
15:12	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
11:0	DWord Length <table border="1"> <tr> <td>Default Value:</td> <td>0018h Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table>	Default Value:	0018h Excludes DWord (0,1)	Format:	=n Total Length - 2		
Default Value:	0018h Excludes DWord (0,1)						
Format:	=n Total Length - 2						
1	31:12	MFX Indirect Bitstream Object - Base Address (Decoder and Stitch Modes) <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies the 4K-byte aligned memory base address for the read-only indirect data object pointed in the MFD_XXX_BSD_OBJECT command for fetching (reading) the compressed Slice Data. This field is only valid in MPEG2, AVC and VC1 decoder VLD mode.</p>	Format:	GraphicsAddress[31:12]			
	Format:	GraphicsAddress[31:12]					
	11:6	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ						
5:0	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
2	31:16	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Reserved for 64-bit address extension.</p>	Format:	MBZ			
	Format:	MBZ					
15:0	MFX Indirect Bitstream Object - Destination Address (Decoder and Stitch Modes)[47:32] <table border="1"> <tr> <td colspan="2" style="text-align: center;">Description</td> </tr> <tr> <td colspan="2">This field is for the upper range of MFX Indirect Bitstream Object Base Address.</td> </tr> <tr> <td colspan="2">This field is used for 48-bit addressing.</td> </tr> </table>	Description		This field is for the upper range of MFX Indirect Bitstream Object Base Address.		This field is used for 48-bit addressing.	
Description							
This field is for the upper range of MFX Indirect Bitstream Object Base Address.							
This field is used for 48-bit addressing.							
3	31:15	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ						



MFX_IND_OBJ_BASE_ADDR_STATE																		
	<p>14:13 MFX Indirect Bitstream Object - Tiled Resource Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Format:	U2																
	Value	Name	Description															
	0h	TRMODE_NONE	No tiled resource															
1h	TRMODE_TILEYF	4KB tiled resources																
2h	TRMODE_TILEYS	64KB tiled resources																
3h	Reserved																	
<p>12:11 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																	
<p>10:9 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																	
<p>8:7 MFX Indirect Bitstream ObjectBase - Arbitration Priority Control</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U2</td> </tr> </table> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Format:	U2	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority						
Format:	U2																	
Value	Name																	
00b	Highest priority																	
01b	Second highest priority																	
10b	Third highest priority																	
11b	Lowest priority																	
<p>6:1 MFX Indirect Bitstream ObjectBase - Index to Memory Object Control State (MOCS) Tables</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U6</td> </tr> </table> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>	Format:	U6																
Format:	U6																	
<p>0 Reserved</p>																		
<p>4 31:12 MFX Indirect Bitstream Object - Access Upper Bound (Decoder and Stitch Modes)</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>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 is not present, i.e. the Indirect Data Length field of the MFD_XXX_BSD_OBJECT command is set to 0. This field is only valid in MPEG2, AVC, VP8, and VC1 decoder VLD mode.</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Programming Notes</td> </tr> </table>	Format:	GraphicsAddress[31:12]	Programming Notes															
Format:	GraphicsAddress[31:12]																	
Programming Notes																		



MFX_IND_OBJ_BASE_ADDR_STATE		
		For VP8 Encoder , this field is corresponding to MFC Indirect PAK-BSE Object - Access Upper Bound in DW24, DW25 . Please program Indirect bitstream upperbound in this field the same as DW24, DW25.
	11:0	Reserved Format: _____ MBZ
5	31:16	Reserved Format: _____ MBZ Reserved for 64-bit address extension.
	15:0	MFX Indirect Bitstream Object UpperBound (Decoder and Stitch Modes)[47:32] Description This field is for the upper range of MFX Indirect Bitstream Object UpperBound. This field is used for 48-bit addressing. Programming Notes For VP8 Encoder , this field is corresponding to MFC Indirect PAK-BSE Object - Access Upper Bound in DW24, DW25 . Please program Indirect bitstream upperbound in this field the same as DW24, DW25.
6	31:12	MFX Indirect MV Object - Base Address Format: _____ GraphicsAddress[31:12] Specifies the 4K-byte aligned memory base address for the read-only indirect data object pointed in the encoder MFC_AVC_PAK_OBJECT command or the decoder MFD_IT_OBJECT command for fetching the per-MB MV data.This field is only valid in AVC encoder mode or in AVC decoder IT mode
	11:6	Reserved Format: _____ MBZ
	5:0	Reserved Format: _____ MBZ
7	31:16	Reserved Format: _____ MBZ Reserved for 64-bit address extension.
	15:0	MFX Indirect MV Object Base Address [47:32] Description This field is for the upper range of MFX Indirect MV Object Base Address. This field is used for 48-bit addressing.
8	31:15	Reserved Format: _____ MBZ



MFX_IND_OBJ_BASE_ADDR_STATE																
14:13	MFX Indirect MV Object Destination - Tiled Resource Mode Format: U2															
	For Media Surfaces: This field specifies the tiled resource mode.															
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description													
	0h	TRMODE_NONE	No tiled resource													
1h	TRMODE_TILEYF	4KB tiled resources														
2h	TRMODE_TILEYS	64KB tiled resources														
3h	Reserved															
Reserved Format: MBZ																
Reserved Format: MBZ																
8:7	MFX Indirect MV Object - Arbitration Priority Control Format: U2															
	This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.															
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 75%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority					
	Value	Name														
	00b	Highest priority														
01b	Second highest priority															
10b	Third highest priority															
11b	Lowest priority															
Reserved Format: MBZ																
Reserved Format: MBZ																
6:1	MFX Indirect MV Object Desitnation - Index to Memory Object Control State (MOCS) Tables Format: U6															
	The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.															
0	Reserved															
9	MFX Indirect MV Object Access Upper Bound Format: GraphicsAddress[31:12]															
	This field specifies the 4K-byte aligned (exclusive) maximum Graphics Memory address access by the indirect data object in the MFC_AVC_PAK_OBJECT / MFD_IT_OBJECT command for the per-MB MV 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 MV Object Base Address state. Hardware ignores this field if indirect data is not present, i.e. the Indirect Data Length field of the MFC_AVC_PAK_OBJECT / MFD_IT_OBJECT command is set to 0. This field is only valid in AVC encoder mode or in AVC decoder IT mode.															



MFX_IND_OBJ_BASE_ADDR_STATE													
	11:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												
10	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Reserved for 64-bit address extension.</p>	Format:	MBZ									
	Format:	MBZ											
15:0	<p>MFX Indirect MV Object UpperBound [47:32]</p> <table border="1"> <tr> <th colspan="3" style="text-align: center;">Description</th> </tr> <tr> <td colspan="3">This field is for the upper range of MFX Indirect MV Object Base Address.</td> </tr> <tr> <td colspan="3">This field is used for 48-bit addressing.</td> </tr> </table>	Description			This field is for the upper range of MFX Indirect MV Object Base Address.			This field is used for 48-bit addressing.					
Description													
This field is for the upper range of MFX Indirect MV Object Base Address.													
This field is used for 48-bit addressing.													
11	31:12	<p>MFD Indirect IT-COEFF Object - Base Address (Decoder Only)</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies the 4K-byte aligned memory base address for the read-only indirect data object pointed in the MFD_IT_OBJECT command for fetching (reading) the per-MB non-scaled coefficient data (all inverse scaling and quantization are done in hardware). This field is only valid in MPEG2, AVC and VC1 decoder IT mode.</p>	Format:	GraphicsAddress[31:12]									
	Format:	GraphicsAddress[31:12]											
	11:6	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												
5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
12	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Reserved for 64-bit address extension.</p>	Format:	MBZ									
	Format:	MBZ											
15:0	<p>MFD Indirect IT-COEFF Object Base Address [47:32]</p> <table border="1"> <tr> <th colspan="3" style="text-align: center;">Description</th> </tr> <tr> <td colspan="3">This field is for the upper range of MFX Indirect IT-COEFF Object Base Address.</td> </tr> <tr> <td colspan="3">This field is for the upper range of MFX Indirect MV Object Base Address.</td> </tr> <tr> <td colspan="3">This field is used for 48-bit addressing.</td> </tr> </table>	Description			This field is for the upper range of MFX Indirect IT-COEFF Object Base Address.			This field is for the upper range of MFX Indirect MV Object Base Address.			This field is used for 48-bit addressing.		
Description													
This field is for the upper range of MFX Indirect IT-COEFF Object Base Address.													
This field is for the upper range of MFX Indirect MV Object Base Address.													
This field is used for 48-bit addressing.													
13	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ											
14:13	<p>MFD Indirect IT-COEFF - Tiled Resource Mode</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource				
Format:	U2												
Value	Name	Description											
0h	TRMODE_NONE	No tiled resource											



MFX_IND_OBJ_BASE_ADDR_STATE												
	1h	TRMODE_TILEYF 4KB tiled resources										
	2h	TRMODE_TILEYS 64KB tiled resources										
	3h	Reserved										
	12:11	Reserved Format: _____ MBZ										
	10:9	Reserved Format: _____ MBZ										
	8:7	MFD Indirect IT-COEFF Object Desitnation - Arbitration Priority Control Format: _____ U2 This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
Value	Name											
00b	Highest priority											
01b	Second highest priority											
10b	Third highest priority											
11b	Lowest priority											
	6:1	MFD Indirect IT-COEFF Object Desitnation - Index to Memory Object Control State (MOCS) Tables Format: _____ U6 The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.										
	0	Reserved										
14	31:12	MFD Indirect IT-COEFF Object - Access Upper Bound (Decoder Only) Format: _____ GraphicsAddress[31:12] This field specifies the 4K-byte aligned (exclusive) maximum Graphics Memory address access by the indirect data object in the MFD_IT_OBJECT command for the per-MB non-scaled coefficient 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 MFD Indirect IT-COEFF Object Base Address state.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 field is only valid in MPEG2, AVC and VC1 decoder IT mode.										
	11:0	Reserved Format: _____ MBZ										
15	31:16	Reserved Format: _____ MBZ Reserved for 64-bit address extension.										



MFX_IND_OBJ_BASE_ADDR_STATE																	
	15:0	MFD Indirect IT-COEFF Object UpperBound [47:32]															
		Description															
		This field is for the upper range of MFX Indirect IT-COEFF Object UpperBound.															
		This field is used for 48-bit addressing.															
16	31:12	MFD Indirect IT-DBLK Object - Base Address (Decoder Only) Format: GraphicsAddress[31:12] Specifies the 4K-byte aligned memory base address for the read-only indirect data object pointed in the MFD_IT_OBJECT command for fetching (reading) the per-MB Deblocking filter control data. This field is only valid in AVC decoder IT mode.															
	11:6	Reserved Format: MBZ															
	5:0	Reserved Format: MBZ															
17	31:16	Reserved Format: MBZ Reserved for 64-bit address extension.															
	15:0	MFD Indirect IT-DBLK Object Base Address [47:32] Description This field is for the upper range of MFX Indirect IT-DBLK Object Base Address. This field is used for 48-bit addressing.															
18	31:15	Reserved Format: MBZ															
	14:13	MFD Indirect IT-DBLK Object Destination - Tiled Resource Mode Format: U2 For Media Surfaces: This field specifies the tiled resource mode. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
	0h	TRMODE_NONE	No tiled resource														
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
12:11	Reserved Format: MBZ																
10:9	Reserved Format: MBZ																



MFX_IND_OBJ_BASE_ADDR_STATE														
	8:7	<p>MFD Indirect IT-DBLK Object - Arbitration Priority Control</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Format:	U2	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
	Format:	U2												
	Value	Name												
	00b	Highest priority												
	01b	Second highest priority												
	10b	Third highest priority												
11b	Lowest priority													
6:1	<p>MFD Indirect IT-DBLK Object Desitnation - Index to Memory Object Control State (MOCS) Tables</p> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>	Format:	U6											
Format:	U6													
0	Reserved													
19	31:12	<p>MFD Indirect IT-DBLK Object - Access Upper Bound (Decoder Only)</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>This field specifies the 4K-byte aligned (exclusive) maximum Graphics Memory address access by the indirect data object in the MFD_IT_OBJECT command for the per-MB Deblocking filter control 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 MFD Indirect IT-DBLK Object Base Address state.Hardware ignores this field if indirect data is not present, i.e. the Indirect Deblocking Control Data Length field of the MFD_IT_OBJECT command is set to 0.This field is only valid in AVC decoder IT mode.</p>	Format:	GraphicsAddress[31:12]										
	Format:	GraphicsAddress[31:12]												
11:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ											
Format:	MBZ													
20	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Reserved for 64-bit address extension.</p>	Format:	MBZ										
	Format:	MBZ												
	15:0	<p>MFD Indirect IT-DBLK Object UpperBound [47:32]</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>This field is for the upper range of MFX Indirect IT-DBLK Object UpperBound.</td> </tr> <tr> <td>This field is used for 48-bit addressing.</td> </tr> </tbody> </table>	Description	This field is for the upper range of MFX Indirect IT-DBLK Object UpperBound.	This field is used for 48-bit addressing.									
Description														
This field is for the upper range of MFX Indirect IT-DBLK Object UpperBound.														
This field is used for 48-bit addressing.														



MFX_IND_OBJ_BASE_ADDR_STATE																			
21	31:12	<p>MFC Indirect PAK-BSE Object - Base Address (Encoder Only)</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies the 4K-byte aligned memory base address for the write-only indirect data object pointed in the PAK_SLICE_STATE command for writing out the compressed bitstream. This field is only valid in AVC encoder mode.</p>	Format:	GraphicsAddress[31:12]															
	Format:	GraphicsAddress[31:12]																	
	11:6	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																		
5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																		
22	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Reserved for 64-bit address extension.</p>	Format:	MBZ															
	Format:	MBZ																	
15:0	<p>MFC Indirect PAK-BSE Object Base Address [47:32]</p> <table border="1"> <tr> <th colspan="3" style="text-align: center;">Description</th> </tr> <tr> <td colspan="3">This field is for the upper range of MFX Indirect PAK-BSE Object Base Address.</td> </tr> <tr> <td colspan="3">This field is used for 48-bit addressing.</td> </tr> </table>	Description			This field is for the upper range of MFX Indirect PAK-BSE Object Base Address.			This field is used for 48-bit addressing.											
Description																			
This field is for the upper range of MFX Indirect PAK-BSE Object Base Address.																			
This field is used for 48-bit addressing.																			
23	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ															
	Format:	MBZ																	
	14:13	<p>MFC Indirect PAK-BSE Object Destination - Tiled Resource Mode</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Format:	U2																	
Value	Name	Description																	
0h	TRMODE_NONE	No tiled resource																	
1h	TRMODE_TILEYF	4KB tiled resources																	
2h	TRMODE_TILEYS	64KB tiled resources																	
3h	Reserved																		
12:11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																		
10:9	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																		



MFX_IND_OBJ_BASE_ADDR_STATE												
	8:7	MFC Indirect PAK-BSE Object Desitnation - Arbitration Priority Control Format: U2 This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
	Value	Name										
	00b	Highest priority										
	01b	Second highest priority										
	10b	Third highest priority										
	11b	Lowest priority										
6:1	MFC Indirect PAK-BSE Object Desitnation - Index to Memory Object Control State (MOCS) Tables Format: U6 The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.											
0	Reserved											
24	31:12	MFC Indirect PAK-BSE Object - Access Upper Bound (Eecoder Only) 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 ignored. If 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. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>For VP8 Encoder, this field should be programmed the same at both DW4, DW5 MFX Indirect Bitstream Object - Access Upper Bound as well as DW24, DW25.</td> </tr> </tbody> </table>	Programming Notes	For VP8 Encoder, this field should be programmed the same at both DW4, DW5 MFX Indirect Bitstream Object - Access Upper Bound as well as DW24, DW25.								
	Programming Notes											
	For VP8 Encoder, this field should be programmed the same at both DW4, DW5 MFX Indirect Bitstream Object - Access Upper Bound as well as DW24, DW25.											
11:0	Reserved Format: MBZ											
25	31:16	Reserved Format: MBZ Reserved for 64-bit address extension.										



MFX_IND_OBJ_BASE_ADDR_STATE						
15:0	<p>MFC Indirect PAK-BSE Object UpperBound [47:32]</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>This field is for the upper range of MFC Indirect PAK-BSE Object UpperBound</td> </tr> <tr> <td>This field is used for 48-bit addressing.</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>For VP8 Encoder, this field should be programmed the same at both DW4, DW5 MFX Indirect Bitstream Object - Access Upper Bound as well as DW24, DW25.</td> </tr> </tbody> </table>	Description	This field is for the upper range of MFC Indirect PAK-BSE Object UpperBound	This field is used for 48-bit addressing.	Programming Notes	For VP8 Encoder, this field should be programmed the same at both DW4, DW5 MFX Indirect Bitstream Object - Access Upper Bound as well as DW24, DW25.
Description						
This field is for the upper range of MFC Indirect PAK-BSE Object UpperBound						
This field is used for 48-bit addressing.						
Programming Notes						
For VP8 Encoder, this field should be programmed the same at both DW4, DW5 MFX Indirect Bitstream Object - Access Upper Bound as well as DW24, DW25.						



MFx_JPEG_HUFF_TABLE_STATE

MFx_JPEG_HUFF_TABLE_STATE						
Source:	VideoCS					
Length Bias:	2					
<p>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.</p>						
DWord	Bit	Description				
0	31:29	Command Type				
		Default Value:	3h PARALLEL_VIDEO_PIPE			
		Format:	OpCode			
	28:27	Pipeline				
		Default Value:	2h MFx_MULTI_DW			
	26:24	Media Command Opcode				
		Default Value:	7h JPEG_COMMON			
	23:21	SubOpcode A				
		Default Value:	0h			
	20:16	SubOpcode B				
Default Value:		2h				
15:12	Reserved					
	Format:	MBZ				
11:0	DWord Length					
	Default Value:	033Dh Excludes DWord (0,1)				
	Format:	=n Total Length - 2				
1	31:1	Reserved				
		Format:	MBZ			
0	0	HuffTableID (1-bit)				
		Identifies the huffman table.				
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Y</td> <td>Huffman table for Y</td> </tr> </tbody> </table>	Value	Name	Description	0
Value	Name	Description				
0	Y	Huffman table for Y				



MFX_JPEG_HUFF_TABLE_STATE				
2..4	31:0	DC_BITS (12 8-bit array) The number of DC Huffman codes of length i, where i is 1~12		
5..7	31:0	DC_HUFFVAL (12 8-bit array) The value associated with each DC Huffman code of length i.		
8..11	31:0	AC_BITS (16 8-bit array) the list of Li, number of Huffman codes of length i, where i is 1~16		
12..51	31:0	AC_HUFFVAL (160 8-bit array) the list of Vi,j, the value associated with each Huffman code of length i		
52	31:16	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
	15:0	AC_HUFFVAL(2-8 bit array) In AC table, BITS can have up to 16-bit codeword. Li can be 0 ~ 162. HUFFVAL will have a list of likely random distributed values		



MFx_JPEG_PIC_STATE

MFx_JPEG_PIC_STATE			
Source:		VideoCS	
Length Bias:		2	
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFx_MULTI_DW
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	7h JPEG
		Format:	OpCode
	23:21	SubOpcode A	
Default Value:		0h Common	
Format:		OpCode	
20:16	SubOpcode B		
	Default Value:	0h MEDIA_	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length	Format:	=n Total Length - 2
	Value	Name	Description
	0001h	[Default]	Excludes DWord (0,1)
1	31	Reserved	
		Exists If:	//Encoder Only
		Format:	MBZ
	30:26	Pixels In Horizontal Last MCU	
Exists If:	//Encoder Only		
The number of pixels in the last MCU in a row MCUs. This information is used for completion of partial MCU.			



MFX_JPEG_PIC_STATE		
31:21	Reserved	
	Exists If:	//Decoder Only
	Format:	MBZ
25:21	Pixels In Vertical Last MCU	
	Exists If:	//Encoder Only
<p>The number of pixels in the last MCU in a column MCUs. This information is used for completion of partial MCU.</p>		
20	Vertical Up-Sampling Enable	
	Exists If:	//Decoder Only
<p>Only applied to chroma blocks. This flag is used for 2:1 vertical up-sampling for chroma 420 and outputting chroma422 YUY2 or UYVY format. To enable this flag, the input should be interleaved Scan, InputFormatYUV should be set to YUV420, and OutputFormatYUV should be set to YUY2 or UYVY.</p>		
	Value	Name Description
	0b	no up-sampling
	1b	2:1 vertical up-sampling
19	Reserved	
	Exists If:	//Decoder Only
18	Horizontal Down-Sampling Enable	
	Exists If:	//Decoder Only
<p>Only applied to chroma blocks. This flag is used for 2:1 horizontal down-sampling for chroma 422 and outputting chroma420 NV21 format. To enable this flag, the input should be interleaved Scan, InputFormatYUV should be set to YUV422V_2Y or YUV422V_4Y, and OutputFormatYUV should be set to NV12.</p>		
	Value	Name Description
	0b	no down-sampling
	1b	2:1 horizontal down-sampling
17	Vertical Down-Sampling Enable	
	Exists If:	//Decoder Only
<p>Only applied to chroma blocks. This flag is used for 2:1 vertical down-sampling for chroma 422 and outputting chroma420 NV21 format. To enable this flag, the input should be interleaved Scan, InputFormatYUV should be set to YUV422H_2Y or YUV422H_4Y, and OutputFormatYUV should be set to NV12.</p>		
	Value	Name Description
	0b	no down-sampling
	1b	2:1 vertical down-sampling

**MFX_JPEG_PIC_STATE**

16	Average Down Sampling	
	Exists If:	//Decoder Only
This flag is used to select a down-sampling method when VertDownSamplingEnb or HoriDownSamplingEnb is set to 1.		
	Value	Name Description
	0b	Drop every other line (or column) pixels
	1b	Average neighboring two pixels
20:12	Reserved	
	Exists If:	//Encoder Only
	Format:	MBZ
15:12	Reserved	
	Exists If:	//Decoder Only
	Format:	MBZ
11:8	Output Format YUV	
	Exists If:	//Decoder Only
This field specifies the surface format to write the decoded JPEG image. Note that any non-interleaved JPEG input should be set to "0000". For the interleaved input Scan data, it can be set either "0000" or the corresponding format.		
	Value	Name Description
	0000b	3 separate plane for Y, U, and V respectively
	0001b	NV12 for chroma 4:2:0
	0010b	UYVY for chroma 4:2:2
	0011b	YUY2 for chroma 4:2:2
Programming Notes		
The MFX_SURFACE_STATE command should be set accordingly for each OutputFormatYUV . For NV12, Surface Format = 4 (PLANAR_420_8) For YUY2, Surface Format = 0 (YCRCB_NORMAL) For UYVY, Surface Format = 3 (YCRCB_SWAPY) NV12 (0001b) can be set only when Y, U, V are interleaved in a single Scan data with the following cases		
<ul style="list-style-type: none"> • InputFormatYUV is YUV420 and VertDownSamplingEnb is disabled • InputFormatYUV is YUV422H_2Y or YUV422H_4Y, and VertDownSamplingEnb is enabled 		
UYVY (0010b) and YUY2 (0011b) can be set only when Y, U, V are interleaved in a single Scan data with the following cases		
<ul style="list-style-type: none"> • InputFormatYUV is YUV420 and VertUpSamplingEnb is enabled • InputFormatYUV is YUV422H_2Y or YUV422H_4Y and VertUpSamplingEnb is disabled 		



MFX_JPEG_PIC_STATE			
11:8	Input Surface Format YUV		
	Exists If:	//Encoder Only	
	This field specifies the surface format to read a YUV image data		
	Value	Name	Description
	0000b		Reserved
	0001b	NV12	NV12 for chroma 4:2:0
	0010b	UYVY	UYVY for chroma 4:2:2
	0011b	YUY2	YUY2 for chroma 4:2:2
	0100b	Y8	Y8 for chroma400 Y-only image
	0101b	RGB	RGB or YUV for chroma 4:4:4
Programming Notes			
This field should be set accordingly for SurfaceFormat in MFX_SURFACE_STATE command.			
R8G8B8A8_UNORM in this field is used for encoding RGB and YUV chroma 4:4:4.			
For RGB input, any order of image components R, G, B (e.g., RGB, GBR, BGR, YUV) will be acceptable as far as the order of Quantization tables and Huffman tables match the order of image components.			
7:6	Reserved		
	Exists If:	//Decoder Only	
	Format:	MBZ	
7:6	Reserved		
	Exists If:	//Encoder Only	
	Format:	MBZ	
5:4	Rotation		
	Exists If:	//Decoder Only	
	Value	Name	Description
	00b		no rotation
	01b		rotate clockwise 90 degree
	10b		rotate counter-clockwise 90 degree (same as rotating 270 degree clockwise)
	11b		rotate 180 degree (NOT the same as flipped on the x-axis)
	Programming Notes		
	Rotation can be set to 01b, 10b, or 11b when OutputFormatYUV is set to 0000b. For other OutputFormatYUV, Rotation is not allowed.		



MFX_JPEG_PIC_STATE		
5:3	Reserved	
	Exists If:	//Encoder Only
	Format:	MBZ
3	Reserved	
	Exists If:	//Decoder Only
	Format:	MBZ
2:0	Input Format YUV	
	Exists If:	//Decoder Only
	Format:	U3
	Value	Name
	Description	
	0	[Default] YUV400 (grayscale image)
	1	YUV420
	2	YUV422H_2Y (Horizontally chroma 2:1 subsampled) - horizontal 2 Y-block, 1U and 1V
	3	YUV444
	4	YUV411
	5	YUV422V_2Y (Vertically chroma 2:1 subsampled) - vertical 2 Y-blocks, 1U and 1V
	6	YUV422H_4Y - 2x2 Y-blocks, vertical 2U and 2V
	7	YUV422V_4Y - 2x2 Y-blocks, horizontal 2U and 2V
2:0	Output MCU Structure	
	Exists If:	//Encoder Only
	Output MCU Structure(OutputMcuStructure) should be set accordingly for each Input Surface Format YUV(InputSurfaceFormatYUV):	
	<ul style="list-style-type: none"> • If InputSurfaceFormatYUV is set to NV12, OutputMCUStructure is set to YUV420. • If InputSurfaceFormatYUV is set to UYVY or YUY2, OutputMCUStructure is set to YUV422H_2Y. • If InputSurfaceFormatYUV is set to Y8, OutputMCuStructure is set to YUV400. • If InputSurfaceFormatYUV is set to RGB (or GBR, BGR, YUV), OutputMCuStructure is set to RGB. • If InputSurfaceFormatYUV is set to RGB, the order of encoded blocks in MCU will be same as the order of input image components. If the order of input image components is RGB (or GBR, BGR, YUV), then the order of blocks will be RGB (or GBR, BGR, YUV respectively). 	
	Value	Name
	Description	
	0	YUV400
		Grayscale Image



MFX_JPEG_PIC_STATE				
		1	YUV420 Both horizontally and vertically chroma 2:1 subsampled	
		2	YUV422H_2Y Horizontally chroma 2:1 subsampled - horizontal 2 Y-blocks, 1 U and 1 V block	
		3	RGB RGB or YUV444: No subsample	
		4		
		5		
		6		
		7		
2	31:30	Reserved		
		Exists If:	//Decoder Only	
		Format:	MBZ	
	31:29	Reserved		
		Exists If:	//Encoder Only	
		Format:	MBZ	
	29	Output Pixel Normalize		
		Exists If:	//Decoder Only	
		JPEG decoded output pixels for Y and U/V in order to adjust display YUV range.		
		Value	Name	Description
0			No Normalization	
1			Normalize output pixels from [0,255] to [16,235]	
1		Normalize output pixels from [0,255] to [16,239]		
28:16	Frame Height In Blocks Minus 1			
	Exists If:	//Decoder Only		
	Format:	U13-1		
	(The number of blocks in height) - 1. This value is calculated using the number of lines Y and vertical sampling factor of the first component V ₁ in Frame header. See the note following this table. For interleaved components, $((Y + (V_1 * 8 - 1)) / (V_1 * 8)) * V_1 - 1$, where "/" is integer division. For non-interleaved components, $((Y + 7) / 8) - 1$.			
28:16	Frame Height In Blks Minus 1			
	Exists If:	//Encoder Only		
	Format:	U13-1		
	(The number of blocks in height) - 1. This value is calculated using the number of lines Y and vertical sampling factor of the first component V ₁ in Frame header. See the note following this table. For interleaved components: $((Y + (V_1 * 8 - 1)) / (V_1 * 8)) * V_1 - 1$ For non-interleaved components: $((Y + 7) / 8) - 1$			



MFX_JPEG_PIC_STATE		
15:13	Reserved	
	Exists If:	//Decoder Only
	Format:	MBZ
15:13	RoundingQuant	
	Exists If:	//Encoder Only
	Rounding value applied to quantization output	
	Value	Name
		Description
	000b	[Default] 1/2
	001b	(1/2 - 1/128)
	010b	(1/2 + 1/128)
	011b	(1/2 - 1/64)
	100b	(1/2 + 1/64)
	101b	(1/2 - 1/32)
	110b	(1/2 - 1/16)
	111b	(1/2 - 1/8)
12:0	Frame Width In Blocks Minus 1	
	Exists If:	//Decoder Only
	Format:	U13-1
	<p>(The number of blocks in width) - 1. This value is calculated using the number of samples per line X and horizontal sampling factor of the first component H₁ 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, $((X + 7) / 8) - 1$.</p>	
12:0	Frame Width In Blks Minus 1	
	Exists If:	//Encoder Only
	Format:	U13-1
	<p>(The number of blocks in width) - 1. This value is calculated using the number of samples per line X and horizontal sampling factor of the first component H₁ 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: $((X + 7) / 8) - 1$</p>	



MFx_MPEG2_PIC_STATE

MFx_MPEG2_PIC_STATE		
Source:	VideoCS	
Length Bias:	2	
<p>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.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
	Format: OpCode	
	28:27	Pipeline
		Default Value: 2h MFx_MPEG2_PIC_STATE
	Format: OpCode	
	26:24	Media Command Opcode
		Default Value: 3h MPEG2_COMMON
Format: OpCode		
23:21	SubOpcode A	
	Default Value: 0h	
Format: OpCode		
20:16	SubOpcode B	
	Default Value: 0h	
Format: OpCode		
15:12	Reserved	
	Format: MBZ	
11:0	DWord Length	
	Default Value: 0h Excludes DWord (0,1)= 00Bh, used for normal decode and encode mode000h, a special case to provide a dummy image state for stitch mode operation. In this case, fields in DW1 which is part of the dummy image state command are ignored by hardware.	
	Format: =n Total Length - 2	
1	31:28	f_code[1][1]. Used for backward motion vector prediction. See ISO/IEC 13818-2 7.6.3.1 for details
	27:24	f_code[1][0]. Used for backward motion vector prediction. See ISO/IEC 13818-2 7.6.3.1 for details



MFX_MPEG2_PIC_STATE												
23:20	<p>f_code[0][1] Used for forward motion vector prediction. See ISO/IEC 13818-2 7.6.3.1 for details</p>											
19:16	<p>f_code[0][0] Used for forward motion vector prediction. See ISO/IEC 13818-2 7.6.3.1 for details</p>											
15:14	<p>Intra DC Precision</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>See ISO/IEC 13818-2 6.3.10 for details.</p>	Format:	U2									
Format:	U2											
13:12	<p>Picture Structure 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_BOTTOM_FIELD11 = MPEG_FRAME</p>											
11	<p>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</p>											
10	<p>Frame Prediction Frame DCT This field provides constraints on the DCT type and prediction type. It affects the syntax of the bitstream.</p>											
9	<p>Concealment Motion Vector Flag This field indicates if the concealment motion vectors are coded in intra macroblocks. It affects the syntax of the bitstream.</p>											
8	<p>Quantizer Scale Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">MPEG_Q_SCALE_TYPE</td> </tr> </table> <p>This field specifies the quantizer scaling type.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>MPEG_QSCALE_LINEAR</td> </tr> <tr> <td>1h</td> <td></td> <td>D MPEG_QSCALE_NONLINEAR esc</td> </tr> </tbody> </table>	Format:	MPEG_Q_SCALE_TYPE	Value	Name	Description	0h		MPEG_QSCALE_LINEAR	1h		D MPEG_QSCALE_NONLINEAR esc
Format:	MPEG_Q_SCALE_TYPE											
Value	Name	Description										
0h		MPEG_QSCALE_LINEAR										
1h		D MPEG_QSCALE_NONLINEAR esc										
7	<p>Intra VLC Format This field is used by VLD</p>											
6	<p>Scan Order</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">MPEG_INVERSESCAN_TYPE</td> </tr> </table> <p>This field specifies the Inverse Scan method for the DCT-domain coefficients in the</p>	Format:	MPEG_INVERSESCAN_TYPE									
Format:	MPEG_INVERSESCAN_TYPE											



MFX_MPEG2_PIC_STATE																	
		blocks of the current picture.															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>MPEG_ZIGZAG_SCAN</td> </tr> <tr> <td>1h</td> <td></td> <td>MPEG_ALTERNATE_VERTICAL_SCAN</td> </tr> </tbody> </table>	Value	Name	Description	0h		MPEG_ZIGZAG_SCAN	1h		MPEG_ALTERNATE_VERTICAL_SCAN						
Value	Name	Description															
0h		MPEG_ZIGZAG_SCAN															
1h		MPEG_ALTERNATE_VERTICAL_SCAN															
	5:0	Reserved															
2	31	<p>I Slice Concealment Mode</p> <p>Exists If: //Decoder</p> <p>This field controls how MPEG decoder handles MB concealment in I Slice</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Intra Concealment</td> <td>Using Coefficient values to handle MB concealment</td> </tr> <tr> <td>1h</td> <td>Inter Concealment</td> <td>Using Motion Vectors to handle MB concealment</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>If this field is set to "1", driver must provide a valid forward reference picture (both top and bottom Field must be valid)</p>	Value	Name	Description	0h	Intra Concealment	Using Coefficient values to handle MB concealment	1h	Inter Concealment	Using Motion Vectors to handle MB concealment						
Value	Name	Description															
0h	Intra Concealment	Using Coefficient values to handle MB concealment															
1h	Inter Concealment	Using Motion Vectors to handle MB concealment															
	30	<p>Reserved</p> <p>Format: MBZ</p>															
	29:28	<p>P/B Slice Concealment Mode</p> <p>Exists If: //Decoder</p> <p>This field controls how MPEG decoder handles MB concealment in P/B Slice.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>INTER</td> <td>If left MB is NOT Intra MB type (including skipMB), use left MB inter prediction mode [frame/field or forward/backward/bi] and MV final values as concealment. Otherwise (left MB is Intra MB), use forward reference (same polarity for field pic) with MV final values set to 0.</td> </tr> <tr> <td>01b</td> <td>LEFT</td> <td>If left MB is NOT Intra MB type (including skipMB), use left MB inter prediction mode [frame/field or forward/backward/bi] and MV final values as concealment. Otherwise (left MB is Intra MB), use left MB dct_dc_pred[cc] values for concealment (Macroblock is concealed as INTRA MB and dct_dc_pred[cc] are DC predictor for Luma, Cr, Cb data)</td> </tr> <tr> <td>10b</td> <td>ZERO</td> <td>Always use forward reference (same polarity for field pic) with MV final values set to 0 (Macroblock is concealed as INTER coded)</td> </tr> <tr> <td>11b</td> <td>INTRA</td> <td>Use left MB dct_dc_pred[cc] values for concealment (Macroblock is concealed as INTRA MB and dct_dc_pred[cc] are DC predictor for Luma, Cr, Cb data)</td> </tr> </tbody> </table>	Value	Name	Description	00b	INTER	If left MB is NOT Intra MB type (including skipMB), use left MB inter prediction mode [frame/field or forward/backward/bi] and MV final values as concealment. Otherwise (left MB is Intra MB), use forward reference (same polarity for field pic) with MV final values set to 0.	01b	LEFT	If left MB is NOT Intra MB type (including skipMB), use left MB inter prediction mode [frame/field or forward/backward/bi] and MV final values as concealment. Otherwise (left MB is Intra MB), use left MB dct_dc_pred[cc] values for concealment (Macroblock is concealed as INTRA MB and dct_dc_pred[cc] are DC predictor for Luma, Cr, Cb data)	10b	ZERO	Always use forward reference (same polarity for field pic) with MV final values set to 0 (Macroblock is concealed as INTER coded)	11b	INTRA	Use left MB dct_dc_pred[cc] values for concealment (Macroblock is concealed as INTRA MB and dct_dc_pred[cc] are DC predictor for Luma, Cr, Cb data)
Value	Name	Description															
00b	INTER	If left MB is NOT Intra MB type (including skipMB), use left MB inter prediction mode [frame/field or forward/backward/bi] and MV final values as concealment. Otherwise (left MB is Intra MB), use forward reference (same polarity for field pic) with MV final values set to 0.															
01b	LEFT	If left MB is NOT Intra MB type (including skipMB), use left MB inter prediction mode [frame/field or forward/backward/bi] and MV final values as concealment. Otherwise (left MB is Intra MB), use left MB dct_dc_pred[cc] values for concealment (Macroblock is concealed as INTRA MB and dct_dc_pred[cc] are DC predictor for Luma, Cr, Cb data)															
10b	ZERO	Always use forward reference (same polarity for field pic) with MV final values set to 0 (Macroblock is concealed as INTER coded)															
11b	INTRA	Use left MB dct_dc_pred[cc] values for concealment (Macroblock is concealed as INTRA MB and dct_dc_pred[cc] are DC predictor for Luma, Cr, Cb data)															
	27	<p>Reserved</p> <p>Format: MBZ</p>															



MFX_MPEG2_PIC_STATE			
26:25	P/B Slice Predicted BiDir Motion Type Override - Bi-direction MV Type Override		
	Exists If: //Decoder		
	This field is only applicable if the Concealment Motion Type is predicted to be Bi-directional. (It is only possible if "P/B Slice Concealment Mode" is set to "00" or "01" and left MB is a bi-directional MB).		
	Value	Name Description	
	0h	BID Keep Bi-direction Prediction	
	1h	RESERVED	
	2h	FWD Only use Forward Prediction (Backward MV is forced to invalid	
	3h	BWD Only use Backward Prediction (Forward MV is forced to invalid)	
	24	P/B Slice Predicted Motion Vector Override Final MV value Override	
		Exists If: //Decoder	
This field is only applicable if the Concealment Motion Vectors are non-zero. It is only possible if "P/B Slice Concealment Mode" is set to "00" or "01" and left MB has non-zero motion vectors).			
Value		Name Description	
0h		Predicted Motion Vectors use predicted values	
1h	ZERO Motion Vectors force to 0		
23:15	Reserved		
	Format: MBZ		
14	LoadSlicePointerFlag - LoadBitStreamPointerPerSlice		
	Exists If: //Encoder		
	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	Name Description	
	0h	Load BitStream Pointer only once for the first slice of a frame	
1h	Load/reload BitStream Pointer only once for the each slice, reload the start location of the bitstream buffer from the Indirect PAK-BSE Object Data Start Address field		
13	Reserved		
	Format: MBZ		
12	Reserved		
	Format: MBZ		



MFX_MPEG2_PIC_STATE																
	11	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
	10:9	Picture Coding Type <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>MPEG_PICTURE_CODING_TYPE</td> </tr> </table> <p>This field identifies whether the picture is an intra-coded picture (I), predictive-coded picture (P) or bi-directionally predictive-coded picture (B). See ISO/IEC 13818-2 6.3.9 for details.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Reserved</td> </tr> <tr> <td>01b</td> <td>MPEG_I_PICTURE</td> </tr> <tr> <td>10b</td> <td>10 = MPEG_P_PICTURE</td> </tr> <tr> <td>11b</td> <td>MPEG_B_PICTURE</td> </tr> </tbody> </table>	Format:	MPEG_PICTURE_CODING_TYPE	Value	Name	00b	Reserved	01b	MPEG_I_PICTURE	10b	10 = MPEG_P_PICTURE	11b	MPEG_B_PICTURE		
	Format:	MPEG_PICTURE_CODING_TYPE														
	Value	Name														
00b	Reserved															
01b	MPEG_I_PICTURE															
10b	10 = MPEG_P_PICTURE															
11b	MPEG_B_PICTURE															
8:2	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ															
1	MismatchControlDisabled These 2 bits flag disables mismatch control of the inverse transformation for some specific cases during reference reconstruction. <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td></td> <td>Mismatch control applies to all MBs</td> </tr> <tr> <td>01b</td> <td></td> <td>Disable mismatch control to all intra MBs whose all AC-coefficients are zero.</td> </tr> <tr> <td>10b</td> <td></td> <td>Disable mismatch control to all MBs whose all AC-coefficients are zero.</td> </tr> <tr> <td>11b</td> <td></td> <td>Disable mismatch control to all MBs.</td> </tr> </tbody> </table>	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 mismatch control to all MBs whose all AC-coefficients are zero.	11b		Disable mismatch control to all MBs.
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 mismatch control to all MBs whose all AC-coefficients are zero.														
11b		Disable mismatch control to all MBs.														
0	Disable Mismatch To disable MPEG2 IDCT fixed point arithmetic correction															
3	31 Slice Concealment Disable Bit <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Exists If:</td> <td>//Decode</td> </tr> </table> <p>If VINunit detects the next slice starting position is either out-of-bound or smaller than or equal to the current slice starting position, VIN will set the current slice to be 1 MB and force VMDunit to do slice concealment on the next slice. This bit will disable this feature and the MB data from the next slice will be decoded from bitstream.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Enable [Default]</td> <td>VIN will force next slice to be concealment if detects slice boundary error</td> </tr> <tr> <td>1h</td> <td>Disable</td> <td>VIN will not force next slice to be in concealment</td> </tr> </tbody> </table> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Programming Notes</td> </tr> </table>	Exists If:	//Decode	Value	Name	Description	0h	Enable [Default]	VIN will force next slice to be concealment if detects slice boundary error	1h	Disable	VIN will not force next slice to be in concealment	Programming Notes			
Exists If:	//Decode															
Value	Name	Description														
0h	Enable [Default]	VIN will force next slice to be concealment if detects slice boundary error														
1h	Disable	VIN will not force next slice to be in concealment														
Programming Notes																



MFX_MPEG2_PIC_STATE																				
		Driver has an option to detect the scenario given in description (above) and remove the second (out-of-order) slice. In this case, hardware will decode the first slice in completion and do concealment till the third slice. It should yield a picture with better quality this way.																		
	30:29	Reserved Format: MBZ																		
	28:24	Reserved																		
	23:16	FrameHeightInMBsMinus1[7:0] (Picture Height in Macroblocks) Format: U8																		
	15:8	Reserved Format: MBZ for future supporting width > 4K																		
	7:0	FrameWidthInMBsMinus1[7:0] (Picture Width in Macroblocks) Format: U8																		
4	31:16	<p>MinFrameWSize</p> <p>Format: U16</p> <p>- Minimum Frame Size [15:0] (16-bit) (Encoder Only) Minimum Frame Size is specified to compensate for intel Rate Control Currently zero fill (no need to perform emulation byte insertion) is done only to the end of the CABAC_ZERO_WORD insertion (if any) at the last slice of a picture. Intel encoder parameter, not part of DXVA. The caller should always make sure that the value, represented by Minimum Frame Size, is always less than maximum frame size FrameBitRateMax (DWORD 10 bits 29:16). This field is reserved in Decode mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,0003FFFFh]</td> <td></td> <td>The programmable range when MinFrameWSizeUnits is 00.</td> </tr> <tr> <td>[0,000FFFFFFh]</td> <td></td> <td>The Programmable range when MinFrameWSizeUnits is 01.</td> </tr> <tr> <td>[0,03FFFFFFh]</td> <td></td> <td>The Programmable range when MinFrameWSizeUnits is 10.</td> </tr> <tr> <td>[0, FFFFFFFFh]</td> <td></td> <td>The Programmable range when MinFrameWSizeUnits is 11.</td> </tr> <tr> <td>0h</td> <td>[Default]</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	[0,0003FFFFh]		The programmable range when MinFrameWSizeUnits is 00.	[0,000FFFFFFh]		The Programmable range when MinFrameWSizeUnits is 01.	[0,03FFFFFFh]		The Programmable range when MinFrameWSizeUnits is 10.	[0, FFFFFFFFh]		The Programmable range when MinFrameWSizeUnits is 11.	0h	[Default]	
Value	Name	Description																		
[0,0003FFFFh]		The programmable range when MinFrameWSizeUnits is 00.																		
[0,000FFFFFFh]		The Programmable range when MinFrameWSizeUnits is 01.																		
[0,03FFFFFFh]		The Programmable range when MinFrameWSizeUnits is 10.																		
[0, FFFFFFFFh]		The Programmable range when MinFrameWSizeUnits is 11.																		
0h	[Default]																			
	15	Reserved Format: MBZ																		
	14:12	RoundInterAC, rounding precision for non-Intra AC000: +1/16001: +2/16010: +3/16011: +4/16100: +5/16101: +6/16110: +7/16111: +8/16																		
	11	Reserved Format: MBZ																		



MFX_MPEG2_PIC_STATE													
	10:8	RoundIntraAC Format: U3 rounding precision for Intra AC000: +1/16001: +2/16010: +3/16011: +4/16100: +5/16101: +6/16110: +7/16111: +8/16											
	7	Reserved Format: MBZ											
	6:4	RoundInterDC rounding Precision for non-Intra-DC000: +1/16001: +2/16010: +3/16011: +4/16100: +5/16101: +6/16110: +7/16111: +8/16											
	3	Reserved Format: MBZ											
	2:1	RoundIntraDC rounding Precision for Intra-DC00: +1/801: +2/810: +3/811: +4/8											
	0	Reserved											
5	31:17	Reserved (for future Mask bits)											
	16	FrameSizeControlMask Frame size conformance maskThis field is used when MacroblockStatEnable is set to 1. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Do not change Slice Quantization Parameter values in MFC_MPEG2_SLICEGROUP_STATE with suggested slice QP value for frame level Rate control</td> </tr> <tr> <td>1h</td> <td></td> <td>Replace Slice Quantization Parameter values in MFC_MPEG2_SLICEGROUP_STATE with suggested slice QP value for frame level Rate control values in MFC_IMAGE_STATUS control register.</td> </tr> </tbody> </table>	Value	Name	Description	0h		Do not change Slice Quantization Parameter values in MFC_MPEG2_SLICEGROUP_STATE with suggested slice QP value for frame level Rate control	1h		Replace Slice Quantization Parameter values in MFC_MPEG2_SLICEGROUP_STATE with suggested slice QP value for frame level Rate control values in MFC_IMAGE_STATUS control register.		
	Value	Name	Description										
	0h		Do not change Slice Quantization Parameter values in MFC_MPEG2_SLICEGROUP_STATE with suggested slice QP value for frame level Rate control										
1h		Replace Slice Quantization Parameter values in MFC_MPEG2_SLICEGROUP_STATE with suggested slice QP value for frame level Rate control values in MFC_IMAGE_STATUS control register.											
15:13	Reserved												
12	InterMBForceCBPZeroControlMask Format: U1 Inter MB Force CBP ZERO mask. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0, FFFFFFFFh]</td> <td></td> <td></td> </tr> <tr> <td>0h</td> <td></td> <td>No effect</td> </tr> <tr> <td>1h</td> <td></td> <td>Zero out all A/C coefficients for the inter MB violating Inter Conformance</td> </tr> </tbody> </table>	Value	Name	Description	[0, FFFFFFFFh]			0h		No effect	1h		Zero out all A/C coefficients for the inter MB violating Inter Conformance
Value	Name	Description											
[0, FFFFFFFFh]													
0h		No effect											
1h		Zero out all A/C coefficients for the inter MB violating Inter Conformance											



MFX_MPEG2_PIC_STATE																	
11:10	MinFrameWSizeUnits This field is the Minimum Frame Size Units <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>compatibility mode</td> <td>Minimum Frame Size is in old mode (words, 2bytes)</td> </tr> <tr> <td>01b</td> <td>16 byte</td> <td>Minimum Frame Size is in 16bytes</td> </tr> <tr> <td>10b</td> <td>4Kb</td> <td>Minimum Frame Size is in 4Kbytes</td> </tr> <tr> <td>11b</td> <td>16Kb</td> <td>Minimum Frame Size is in 16Kbytes</td> </tr> </tbody> </table>		Value	Name	Description	00b	compatibility 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
Value	Name	Description															
00b	compatibility 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	MBRateControlMask MB Rate Control conformance mask This field is ignored when MacroblockStatEnable is disabled or MB level Rate control flag for the current MB is disable in Macroblock Status Buffer. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Do not change QP values of inter macroblock with suggested QP values in Macroblock Status Buffer</td> </tr> <tr> <td>1h</td> <td></td> <td>Apply RC QP delta for all macroblock</td> </tr> </tbody> </table>		Value	Name	Description	0h		Do not change QP values of inter macroblock with suggested QP values in Macroblock Status Buffer	1h		Apply RC QP delta for all macroblock						
Value	Name	Description															
0h		Do not change QP values of inter macroblock with suggested QP values in Macroblock Status Buffer															
1h		Apply RC QP delta for all macroblock															
8	Reserved																
7	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ													
Format:	MBZ																
6:4	Reserved																
3	FrameBitRateMinReportMask This is a mask bit controlling if the condition of frame level bit count is less than FrameBitRateMin. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Do not update bit0 of MFC_IMAGE_STATUS control register.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>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.</td> </tr> </tbody> </table>		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.						
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	FrameBitRateMaxReportMask This is a mask bit controlling if the condition of frame level bit count exceeds FrameBitRateMax. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Do not update bit0 of MFC_IMAGE_STATUS control register.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>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.</td> </tr> </tbody> </table>		Value	Name	Description	0h	Disable	Do not update 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.						
Value	Name	Description															
0h	Disable	Do not update 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.															
1	InterMBMaxSizeReportMask This is a mask bit controlling if the condition of any inter MB in the frame exceeds InterMBMaxSize. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> </tbody> </table>		Value	Name	Description												
Value	Name	Description															



MFX_MPEG2_PIC_STATE												
		<table border="1"> <tr> <td>0h</td> <td></td> <td>Do not update bit0 of MFC_IMAGE_STATUS control register.</td> </tr> <tr> <td>1h</td> <td></td> <td>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.</td> </tr> </table>	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 Inter MB Conformance Max size limit.				
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 Inter MB Conformance Max size limit.										
	0	<p>IntraMBMaxSizeReportMask This is a mask bit controlling if the condition of any intra MB in the frame exceeds IntraMBMaxSize.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td></td> <td>Do not update bit0 of MFC_IMAGE_STATUS control register.</td> </tr> <tr> <td>1h</td> <td></td> <td>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.</td> </tr> </tbody> </table>	Value	Name	Description	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.	
Value	Name	Description										
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 [ExistsIf]Encode Only	31:28	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	27:16	<p>InterMBMaxSize</p> <table border="1"> <tr> <td>Default Value:</td> <td>FFFh</td> </tr> </table> <p>This field, Inter MB Conformance Max size limit, indicates the allowed max bit count size for Inter MB</p>	Default Value:	FFFh								
	Default Value:	FFFh										
15:12	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
11:0	<p>IntraMBMaxSize</p> <table border="1"> <tr> <td>Default Value:</td> <td>FFFh</td> </tr> </table> <p>This field, Intra MB Conformance Max size limit, indicates the allowed max bit count size for Intra MB</p>	Default Value:	FFFh									
Default Value:	FFFh											
7	31:1	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	0	<p>VSL top MB Trans8x8flag</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Encode Only</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>VSL will only fetch the current MB data.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>When this bit is set VSL will make extra fetch to memory to fetch the MB data for top MB.</td> </tr> </tbody> </table>	Exists If:	//Encode Only	Value	Name	Description	0	Disable	VSL will only fetch the current MB data.	1	Enable
Exists If:	//Encode Only											
Value	Name	Description										
0	Disable	VSL will only fetch the current MB data.										
1	Enable	When this bit is set VSL will make extra fetch to memory to fetch the MB data for top MB.										



MFX_MPEG2_PIC_STATE										
8 [ExistsIf]Encode Only	31:24	SliceDeltaQPMax[3] <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S7</td> </tr> </table> <p>This field is the Slice level delta QP for total bit-count above FrameBitRateMax - first 1/8 region 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 exceeds FrameBitRateMax but is within 1/8 of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of (FrameBitRateMax, (FrameBitRateMax+ FrameBitRateMaxDelta»3).</p> <p>Range: [-30,30]</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> </tr> <tr> <td>1h</td> <td>Enable</td> </tr> </tbody> </table>	Format:	S7	Value	Name	0h	Disable	1h	Enable
	Format:	S7								
	Value	Name								
	0h	Disable								
1h	Enable									
23:16	SliceDeltaQPMax[2] <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S7</td> </tr> </table> <p>Range: [-30,30]</p> <p>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 ¼ of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta»3), (FrameBitRateMax+ FrameBitRateMaxDelta»2).</p>	Format:	S7							
Format:	S7									
15:8	SliceDeltaQPMax[1] <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S7</td> </tr> </table> <p>Range: [-30,30]</p> <p>This field is the Slice level delta QP for bit-count above FrameBitRateMax - above 1/4 and below 1/2 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 ¼ and ½ of FrameBitRateMaxDelta above FrameBitRateMax, i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta»2), (FrameBitRateMax+ FrameBitRateMaxDelta»1).</p>	Format:	S7							
Format:	S7									
7:0	SliceDeltaQPMax[0] <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S7</td> </tr> </table> <p>Range: [-30,30]</p> <p>This field is the Slice level delta QP for bit-count above FrameBitRateMax - above 1/2 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 above FrameBitRateMax by more than half the distance of FrameBitRateMaxDelta , i.e., in the range of</p>	Format:	S7							
Format:	S7									



MFX_MPEG2_PIC_STATE			
	<p>((FrameBitRateMax+ FrameBitRateMaxDelta»1), infinite).</p>		
<p>9 [ExistsIf]Encode Only</p>	<p>31:24 SliceDeltaQPMIn[3]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S7</td> </tr> </table> <p>Range: [-30,30]</p> <p>This field is the Slice level delta QP for total bit-count below FrameBitRateMin - first 1/8 region 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 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).</p>	Format:	S7
	Format:	S7	
	<p>23:16 SliceDeltaQPMIn[2]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S7</td> </tr> </table> <p>Range: [-30,30]</p> <p>This field is the Slice level delta QP for bit-count below FrameBitRateMin - below 1/ 8 and above 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 one-eighth and quarter the distance of FrameBitRateMinDelta from FrameBitRateMin, i.e., in the range of [(FrameBitRateMin- FrameBitRateMinDelta»2), (FrameBitRateMin- FrameBitRateMinDelta»3)].</p>	Format:	S7
	Format:	S7	
<p>15:8 SliceDeltaQPMIn[1]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S7</td> </tr> </table> <p>Range: [-30,30]</p> <p>This field is the Slice level delta QP for bit-count below FrameBitRateMin- below 1/4 and above 1/ 2 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 quarter and half the distance of FrameBitRateMinDelta from FrameBitRateMin, i.e., in the range of [(FrameBitRateMin- FrameBitRateMinDelta»1), (FrameBitRateMin- FrameBitRateMinDelta»2)].</p>	Format:	S7	
Format:	S7		
<p>7:0 SliceDeltaQPMIn[0]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S7</td> </tr> </table> <p>Range: [-30,30]</p> <p>This field is the Slice Level Delta QP for bit-count below FrameBitRateMin - below 1/ 2 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 below FrameBitRateMin by more than half the distance of FrameBitRateMinDelta , i.e., in the range of [0, (FrameBitRateMin- FrameBitRateMinDelta»1).</p>	Format:	S7	
Format:	S7		



MFX_MPEG2_PIC_STATE											
10 [ExistsIf]Encode Only	31	<p>FrameBitrateMaxUnit This field is the Frame Bitrate Maximum Limit Units.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Byte</td> <td>FrameBitRateMax is in units of 32 Bytes when FrameBitrateMaxUnitMode is 1 and in units of 128 Bytes if FrameBitrateMaxUnitMode is 0</td> </tr> <tr> <td>1h</td> <td>Kilobyte</td> <td>FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0</td> </tr> </tbody> </table>	Value	Name	Description	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
	Value	Name	Description								
	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	<p>FrameBitrateMaxUnitMode BitFiel This field is the Frame Bitrate Maximum Limit Units.dDesc</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Compatibility mode</td> <td>FrameBitRateMaxUnit is in old mode (128b/16Kb)</td> </tr> <tr> <td>1h</td> <td>New mode</td> <td>FrameBitRateMaxUnit is in new mode (32byte/4Kb)</td> </tr> </tbody> </table>	Value	Name	Description	0h	Compatibility mode	FrameBitRateMaxUnit is in old mode (128b/16Kb)	1h	New mode	FrameBitRateMaxUnit is in new mode (32byte/4Kb)
	Value	Name	Description								
	0h	Compatibility mode	FrameBitRateMaxUnit is in old mode (128b/16Kb)								
	1h	New mode	FrameBitRateMaxUnit is in new mode (32byte/4Kb)								
	29:16	<p>FrameBitRateMax This field is the Frame Bitrate Maximum Limit. This field along with FrameBitrateMaxUnit determines maximum allowed bits in a frame before multi-pass gets triggered (when enabled). In other words, multi-pass is triggered when the actual 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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-512KB</td> <td></td> <td>The programmable range 0-512KB when FrameBitrateMaxUnit is 0.</td> </tr> <tr> <td>0-8190KB</td> <td></td> <td>The programmable range 0-8190KB when FrameBitrateMaxUnit is 1.</td> </tr> </tbody> </table>	Value	Name	Description	0-512KB		The programmable range 0-512KB when FrameBitrateMaxUnit is 0.	0-8190KB		The programmable range 0-8190KB when FrameBitrateMaxUnit is 1.
	Value	Name	Description								
0-512KB		The programmable range 0-512KB when FrameBitrateMaxUnit is 0.									
0-8190KB		The programmable range 0-8190KB when FrameBitrateMaxUnit is 1.									
15	<p>FrameBitrateMinUnit This field is the Frame Bitrate Minimum Limit Units.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Byte</td> <td>FrameBitRateMax is in units of 32 Bytes when FrameBitrateMinUnitMode is 1 and in units of 128 Bytes if FrameBitrateMinUnitMode is 0</td> </tr> <tr> <td>1h</td> <td>KiloByte</td> <td>FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0</td> </tr> </tbody> </table>	Value	Name	Description	0h	Byte	FrameBitRateMax is in units of 32 Bytes when FrameBitrateMinUnitMode is 1 and in units of 128 Bytes if FrameBitrateMinUnitMode is 0	1h	KiloByte	FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0	
Value	Name	Description									
0h	Byte	FrameBitRateMax is in units of 32 Bytes when FrameBitrateMinUnitMode is 1 and in units of 128 Bytes if FrameBitrateMinUnitMode is 0									
1h	KiloByte	FrameBitRateMax is in units of 4KBytes Bytes when FrameBitrateMaxUnitMode is 1 and in units of 16KBytes if FrameBitrateMaxUnitMode is 0									
14	<p>FrameBitrateMinUnitMode This field is the Frame Bitrate Minimum Limit Units.ValueNameDescriptionProject</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>compatibility mode</td> <td>FrameBitRateMaxUnit is in old mode (128b/16Kb)</td> </tr> <tr> <td>1h</td> <td>New Mode</td> <td>FrameBitRateMaxUnit is in new mode (32byte/4Kb)</td> </tr> </tbody> </table>	Value	Name	Description	0h	compatibility mode	FrameBitRateMaxUnit is in old mode (128b/16Kb)	1h	New Mode	FrameBitRateMaxUnit is in new mode (32byte/4Kb)	
Value	Name	Description									
0h	compatibility mode	FrameBitRateMaxUnit is in old mode (128b/16Kb)									
1h	New Mode	FrameBitRateMaxUnit is in new mode (32byte/4Kb)									



MFX_MPEG2_PIC_STATE									
	<p>13:0 FrameBitRateMin This field is the Frame Bitrate Minimum Limit ()This field along with FrameBitrateMinUnit determines minimum allowed bits in a Frame before Multi-Pass gets triggered (when enabled). In other words, multi-pass is triggered when the actual frame byte count is less than this value. When FrameBitrateMinUnitMode is 0 (compatibility mode) bits 0:11 should be used, bits 12 and 13 should be 0. Range: The programmable range 0-512KB When FrameBitrateMinUnit is in 0. Programmable range is 0-8190 KB when FrameBitrateMinUnit is in 1</p>								
11 [ExistsIf]Encode Only	<p>31 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
	Format:	MBZ							
	<p>30:16 FrameBitRateMaxDelta</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>0h</td> </tr> <tr> <td>Access:</td> <td>None</td> </tr> <tr> <td>Format:</td> <td>U15</td> </tr> </table> <p>This field is used to select the slice delta QP when FrameBitRateMax Is exceeded. It shares the same FrameBitrateMaxUnit. The programmable range is either 0- 512KB or 4MBB in FrameBitrateMaxUnit of 128 Bytes or 16KB respectively.</p> <p>This field is used to select the slice delta QP when FrameBitRateMax Is exceeded. It shares the same FrameBitrateMaxUnit. When FrameBitrateMaxUnitMode is 0(compatibility mode) bits 16:27 should be used, bits 28, 29 and 30 should be 0.</p>	Default Value:	0h	Access:	None	Format:	U15		
	Default Value:	0h							
Access:	None								
Format:	U15								
<p>15 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ								
<p>14:0 FrameBitRateMinDelta This field is used to select the slice delta QP when FrameBitRateMin Is exceeded. It shares the same FrameBitrateMinUnit. When FrameBitrateMinUnitMode is 0(compatibility mode) bits 0:11 should be used, bits 12, 13 and 14 should be 0.Note: HW requires the following condition $FrameBitRateMinDelta \leq 2 * FrameBitRateMin$ Must be true, otherwise it may cause unpredicted behavior.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0-1024KB</td> <td></td> <td>The programmable range 0-1024KB When FrameBitrateMinUnit is in 32Bytes.</td> </tr> <tr> <td>0-16380KB</td> <td></td> <td>Programmable range is 0-16380KB when FrameBitrateMinUnit is in 4Kbytes.</td> </tr> </tbody> </table>	Value	Name	Description	0-1024KB		The programmable range 0-1024KB When FrameBitrateMinUnit is in 32Bytes.	0-16380KB		Programmable range is 0-16380KB when FrameBitrateMinUnit is in 4Kbytes.
Value	Name	Description							
0-1024KB		The programmable range 0-1024KB When FrameBitrateMinUnit is in 32Bytes.							
0-16380KB		Programmable range is 0-16380KB when FrameBitrateMinUnit is in 4Kbytes.							
12	<p>31:21 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ								



MFX_MPEG2_PIC_STATE				
20	VMD Error Logic			
	Value	Name	Description	
	0	Disable [Default]		
	1	Enable	Error Handling	
	Reserved			
	Format:		MBZ	
	18	VAD Error Logic		
		Value	Name	Description
		0	Enable [Default]	Error reporting ON in case of premature Slice done
		1	Disable	CABAC Engine will auto decode the bitstream in case of premature slice done.
17	Reserved			
16	RESERVED			
15:0	Reserved			
	Format:		MBZ	



MFX_PAK_INSERT_OBJECT

MFX_PAK_INSERT_OBJECT	
Source:	VideoCS
Length Bias:	2
Description	
<p>The MFX_PAK_INSERT_OBJECT command is the first primitive command for the AVC, MPEG2, JPEG, and VP8 Encoding Pipeline.</p> <p>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 to perform the actual insertion by transferring the command inline data to the output buffer max, 32 bits at a time.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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).</p> <p>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.</p> <p>Insertion data can include: any encoded syntax elements bit data before the encoded Slice Data (PAK Object Command) of the current Slice SPS NAL PPS NAL SEI NAL Other Non-Slice NAL Leading_Zero_8_bits (as many bytes as there is) Start Code Prefix NAL Header Byte Slice Header Any 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 bitstream, whichever comes first Cabac_Zero_Word or Trailing_Zero_8bits (as many bytes as there is).</p> <p>Anything listed above before a Slice Data Context switch interrupt is not supported by this command.</p>	



DWord	Bit	Description								
0	31:29	Command Type								
		Default Value: 3h PARALLEL_VIDEO_PIPE Format: OpCode								
	28:27	Pipeline								
		Default Value: 2h MFX_PAK_INSERT_OBJECT Format: OpCode								
	26:24	Media Command Opcode								
		Default Value: 0h MFX_COMMON 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									
	Default Value: 0h Excludes DWord (0,1) = Variable Length in DW Format: =n Total Length - 2									
1	31:18	Reserved Format: MBZ								
	17:16	DataByteOffset - SrcDataStartingByteOffset[1:0] Source Data Starting Byte Position within the very first inline DW. Programming Notes Must be set to 0 for JPEG encoder								
	15	HeaderLengthExcludeFrmSize In case this flag is on, bits are NOT accumulated during current access unit coding neither for Cabac Zero Word insertion bits counting or for output in MMIO register MFC_BITSTREAM_BYTECOUNT_FRAME_NO_HEADER. When using HeaderLengthExcludeFrmSize for header insertion, the software needs to make sure that data comes already with inserted start code emulation bytes. SW shouldn't set EmulationFlag bit (Bit 3 of DWORD1 of MFX_PAK_INSERT_OBJECT). <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>NO_ACCUMULATION</td> <td>Bits during current call are not accumulated</td> </tr> <tr> <td>0</td> <td>ACCUMULATE</td> <td>All bits accumulated</td> </tr> </tbody> </table>	Value	Name	Description	1	NO_ACCUMULATION	Bits during current call are not accumulated	0	ACCUMULATE
Value	Name	Description								
1	NO_ACCUMULATION	Bits during current call are not accumulated								
0	ACCUMULATE	All bits accumulated								



MFX_PAK_INSERT_OBJECT		
Programming Notes		
Must be set to 0 for JPEG encoder		
14	Slice Header Indicator This bit indicates if the insert object is a slice header.	
	Value	Name
	1	SLICE_HEADER
	0	LEGACY
		Description
		Insertion Object is a Slice Header. The command is stored internally by HW and is used for inserting slice headers.
		Legacy Insertion Object command. The PAK Insertion Object command is not stored in HW.
Programming Notes		
The payload for PAK_INS_OBJ should contain only start code for Slice header followed by NAL_type and slice header (slice_header() in AVC spec). The payload for PAK_INS_OBJ shouldn't contain 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.		
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.	
	Value	Name
	[1,32]	
7:4	SkipEmulByteCnt - Skip Emulation Byte Count Skip emulation check for number of starting bytes. It can be programmed from 0 to 15 bytes. For example, to skip the start code that has already prefixed in the bitstream.	
Programming Notes		
Must be set to 0 for JPEG encoder		
3	EmulationFlag - EmulationByteBitsInsertEnable	
	Value	Name
	0	NONE
	1	EMULATE
		Description
		No emulation
		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.
Programming Notes		



MFX_PAK_INSERT_OBJECT											
		Must be set to 0 for JPEG encoder									
	2	<p>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 CAVLC, hardware ignores this bit</p>									
	1	<p>EndOfSliceFlag - LastDstDataInsertCommandFlag No more insertion command and no more PAK-OBJECT command follows. Flush data out to memory</p>									
	0	<p>BitstreamStartReset - ResetBitStreamStartingPos</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>RESET</td> <td>Reset the bitstream buffer insertion position to the bitstream buffer starting position.</td> </tr> <tr> <td>0</td> <td>INSERT</td> <td>Insert the current command inline data starting at the current bitstream buffer insertion position</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Must be set to 1 for JPEG encoder</p>	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
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									
2..n	31:0	<p>Insert Data PayLoad Actual Data to be inserted to the output bitstream buffer.</p>									



MFX_PIPE_BUF_ADDR_STATE

MFX_PIPE_BUF_ADDR_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>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).</p> <p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_PIPE_BUF_ADDR_STATE
		Format:	OpCode
	26:24	Common Opcode	
		Default Value:	0h MFX_COMMON_STATE
		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		
	Format:	=n	
	Fixed Length		
	Value	Name	Description
3Fh	DWORD_COUNT_n [Default]	Excludes DWord (0,1)	



MFX_PIPE_BUF_ADDR_STATE																	
1	31:6	<p>Pre Deblocking Destination Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Specifies the 4K byte aligned frame buffer address for outputting the non-filtered reconstructed YUV picture (i.e. output of final adder in each codec standard, and prior to the deblocking filter unit). This field is ignored if PreDeblockOutEnable is set to 0 (disable).</p>	Format:	GraphicsAddress[31:6]													
	Format:	GraphicsAddress[31:6]															
5:0	Reserved																
2	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
15:0	<p>Pre Deblocking Destination Address High</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Pre-Deblocking Destination Address. This field is ignored if PreDeblockOutEnable is set to 0 (disable).</p>	Format:	GraphicsAddress[47:32]														
Format:	GraphicsAddress[47:32]																
3	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	<p>Pre Deblocking - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
0h	TRMODE_NONE	No tiled resource															
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
12:11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
10	<p>Pre Deblocking - Memory Compression Mode</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e1eef6;">Description</th> </tr> <tr> <td colspan="2">Distinguishes Vertical from Horizontal compression. Please refer to vol1aMemory Data Formats chapter -sectionMedia Memory Compressionfor more details.</td> </tr> <tr> <th style="width: 15%;">Value</th> <th style="width: 85%;">Name</th> </tr> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </table> <table border="1" style="width: 100%;"> <tr> <th style="text-align: center; background-color: #e1eef6;">Programming Notes</th> </tr> <tr> <td>This bit is not used unless Memory Compression Enable is set to "1"</td> </tr> <tr> <td>All Codec except JPEG Vertical Compression Only</td> </tr> </table>	Description		Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details.		Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode	Programming Notes	This bit is not used unless Memory Compression Enable is set to "1"	All Codec except JPEG Vertical Compression Only			
Description																	
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details.																	
Value	Name																
0	Horizontal Compression Mode																
1	Vertical Compression Mode																
Programming Notes																	
This bit is not used unless Memory Compression Enable is set to "1"																	
All Codec except JPEG Vertical Compression Only																	



MFX_PIPE_BUF_ADDR_STATE

Video Mode	Compression Mode
AVC Frame Only (No MBAFF or Field)	Vertical
VP8 (Only Frame is supported)	Vertical

JPEG Decode Table

Chroma Format	Output Format	Compression Mode
422H_2Y, 422H_4Y	YUY2	Horizontal
422H_2Y, 422H_4Y	UYVY	Horizontal
422H_2Y, 422H_4Y	NV12	Not Supported
420	NV12	Vertical

9 Pre Deblocking - Memory Compression Enable

Format:	Enable
---------	--------

Memory compression will be attempted for this surface.

Value	Name
0	Compression Disable
1	Compression Enable

Programming Notes

Video Mode	Compression Enable
AVC Frame Only (No MBAFF or Field)	Yes
VP8 (Only Frame is supported)	Yes

JPEG Decode

Chroma Format	Output Format	Compression Enable
422H_2Y,422H_4Y	YUY2	Yes
422H_2Y,422H_4Y	YUY2	Yes
422H_2Y,422H_4Y	UYVY	Yes
422H_2Y, 422H_4Y, 422V_2Y, 422V_4Y	NV12	No
420	YUY2, UYVY	No
420	NV12	Yes

8:7 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 priority
11b	Lowest priority



MFX_PIPE_BUF_ADDR_STATE																	
	6:1	<p>Pre Deblocking - Index to Memory Object Control State (MOCS) Tables: The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>															
	0	Reserved															
4	31:6	<p>Post Deblocking Destination Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Specifies the 4K byte aligned frame buffer address for outputting the post-loop filtered reconstructed YUV picture (i.e. output of the deblocking filter unit) This field is ignored if PostDeblockOutEnable is set to 0 (disable).</p>	Format:	GraphicsAddress[31:6]													
	Format:	GraphicsAddress[31:6]															
5:0	Reserved																
5	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
15:0	<p>Post Deblocking Destination Address High</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Post-Deblocking Destination Address. This field is ignored if PostDeblockOutEnable is set to 0 (disable).</p>	Format:	GraphicsAddress[47:32]														
Format:	GraphicsAddress[47:32]																
6	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	<p>Post Deblocking - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
0h	TRMODE_NONE	No tiled resource															
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
12:11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
10	<p>Post Deblocking - Memory Compression Mode</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> <tr> <td colspan="2">Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details.</td> </tr> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> <tr> <td>0</td> <td>Horizontal Compression Mode</td> </tr> </table>	Description		Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details.		Value	Name	0	Horizontal Compression Mode								
Description																	
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details.																	
Value	Name																
0	Horizontal Compression Mode																



MFX_PIPE_BUF_ADDR_STATE																			
	<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">This bit is not used unless Memory Compression Enable is set to "1"</td> </tr> <tr> <td colspan="2">All Codec except JPEG Vertical Compression Only</td> </tr> <tr> <th style="text-align: center;">Video Mode</th> <th style="text-align: center;">Compression Mode</th> </tr> <tr> <td>AVC Frame Only (No MBAFF or Field)</td> <td>Horizontal</td> </tr> <tr> <td>VP8 (Only Frame is supported)</td> <td>Horizontal</td> </tr> <tr> <td>JPEG</td> <td>Not Supported</td> </tr> </tbody> </table>	Programming Notes		This bit is not used unless Memory Compression Enable is set to "1"		All Codec except JPEG Vertical Compression Only		Video Mode	Compression Mode	AVC Frame Only (No MBAFF or Field)	Horizontal	VP8 (Only Frame is supported)	Horizontal	JPEG	Not Supported				
Programming Notes																			
This bit is not used unless Memory Compression Enable is set to "1"																			
All Codec except JPEG Vertical Compression Only																			
Video Mode	Compression Mode																		
AVC Frame Only (No MBAFF or Field)	Horizontal																		
VP8 (Only Frame is supported)	Horizontal																		
JPEG	Not Supported																		
9	<p>Post Deblocking - Memory Compression Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p>Memory compression will be attempted for this surface.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <th style="text-align: center;">Video Mode</th> <th style="text-align: center;">Compression Enable</th> </tr> <tr> <td>AVC Frame Only (No MBAFF or Field)</td> <td>Yes</td> </tr> <tr> <td>VP8 (Only Frame is supported)</td> <td>Yes</td> </tr> <tr> <td>JPEG</td> <td>No (In JPEG mode, this surface is not used)</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	0	Compression Disable	1	Compression Enable	Programming Notes		Video Mode	Compression Enable	AVC Frame Only (No MBAFF or Field)	Yes	VP8 (Only Frame is supported)	Yes	JPEG	No (In JPEG mode, this surface is not used)
Format:	Enable																		
Value	Name																		
0	Compression Disable																		
1	Compression Enable																		
Programming Notes																			
Video Mode	Compression Enable																		
AVC Frame Only (No MBAFF or Field)	Yes																		
VP8 (Only Frame is supported)	Yes																		
JPEG	No (In JPEG mode, this surface is not used)																		
8:7	<p>Post Deblocking - Arbitration Priority Control</p> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority								
Value	Name																		
00b	Highest priority																		
01b	Second highest priority																		
10b	Third highest priority																		
11b	Lowest priority																		
6:1	<p>Post Deblocking - Index to Memory Object Control State (MOCS) Tables:</p> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.</p> <p>The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>																		
0	Reserved																		
7	<p>31:6 Original Uncompressed Picture Source Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Specifies the 64 byte aligned frame buffer address for fetching YUV pixel data from the original uncompressed input picture for encoding.</p> <p>This field is only valid in encoding mode.</p>	Format:	GraphicsAddress[31:6]																
Format:	GraphicsAddress[31:6]																		



MFX_PIPE_BUF_ADDR_STATE																	
	5:0	Reserved Format: _____ MBZ															
	8	31:16 Reserved Format: _____ MBZ															
	15:0	Original Uncompressed Picture Source Address High Format: _____ GraphicsAddress[47:32] This field is for the upper range of Original Uncompressed Picture Source Address. This field is valid for encoding mode only.															
	9	31:15 Reserved Format: _____ MBZ															
	14:13	Original Uncompressed Picture - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td style="text-align: center;">2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td style="text-align: center;">3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
	0h	TRMODE_NONE	No tiled resource														
	1h	TRMODE_TILEYF	4KB tiled resources														
	2h	TRMODE_TILEYS	64KB tiled resources														
	3h	Reserved															
12:11	Reserved Format: _____ MBZ																
10	Original Uncompressed Picture - Memory Compression Mode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Description		Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before		Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode						
Description																	
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before																	
Value	Name																
0	Horizontal Compression Mode																
1	Vertical Compression Mode																
9	Original Uncompressed Picture - Memory Compression Enable Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable												
Value	Name																
0	Compression Disable																



MFX_PIPE_BUF_ADDR_STATE																
	8:7	<p>Original Uncompressed Picture Source - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority				
	Value	Name														
	00b	Highest priority														
	01b	Second highest priority														
	10b	Third highest priority														
11b	Lowest priority															
6:1	<p>Original Uncompressed Picture - Index to Memory Object Control State (MOCS) Tables The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>															
0	Reserved															
10	31:6	<p>StreamOut Data Destination Base Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Specifies the 64 byte aligned address for outputting the per-MB indirect data to memory when StreamOutEnable is set in the MFX_PIPE_MODE_SELECT command. For Decoder : This field is used for transcoding purpose. 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.</p>	Format:	GraphicsAddress[31:6]												
	Format:	GraphicsAddress[31:6]														
5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ															
11	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
15:0	<p>StreamOut Data Destination Base Address High</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Original Uncompressed Picture Source Address</p>	Format:	GraphicsAddress[47:32]													
Format:	GraphicsAddress[47:32]															
12	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
14:13	<p>StreamOut Data Destination - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
Value	Name	Description														
0h	TRMODE_NONE	No tiled resource														
1h	TRMODE_TILEYF	4KB tiled resources														
2h	TRMODE_TILEYS	64KB tiled resources														
3h	Reserved															



MFX_PIPE_BUF_ADDR_STATE												
	12:11	Reserved										
		Format: MBZ										
	10	StreamOut Data Destination - Memory Compression Mode										
		Description										
		Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before										
		Only horizontal mode is supported.										
		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode				
Value	Name											
0	Horizontal Compression Mode											
1	Vertical Compression Mode											
	9	StreamOut Data Destination - Memory Compression Enable										
		Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.										
		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable						
Value	Name											
0	Compression Disable											
	8:7	StreamOut Data Destination - Arbitration Priority Control										
		This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.										
		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
	Value	Name										
	00b	Highest priority										
01b	Second highest priority											
10b	Third highest priority											
11b	Lowest priority											
	6:1	StreamOut Data Destination - Index to Memory Object Control State (MOCS) Tables										
		The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.										
	0	Reserved										
13	31:6	Intra Row Store Scratch Buffer Base Address										
		Format: GraphicsAddress[31:6]										
		This field provides the base address of the scratch buffer (read/write) used by the AVC/VP8 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.										



MFX_PIPE_BUF_ADDR_STATE																	
		<p>Max 256 cachelines for 4K pixels (1 cacheline for either MBAFF or non-MBAFF)Intra Row Store Scratch Buffer - Arbitration Priority Control</p> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> <p>This is one of the four RowStore Scratch Buffers which can be programmed to use the internal Media Cache (total size 640 CacheLine). When Intra Row Store Scratch Buffer Cache Select is programmed to "1", this data will be stored inside MFX Media Internal Stroage. Driver then needs to program this Base Address between 0 to 639, indicating starting cachelines address to Media Cache. Driver needs to make sure the whole buffer fits into MFX Media Internal Storage. (Notes: 1 cacheline per MB, and the buffer needs to have enough space for 1 MB row).</p> </td> </tr> </tbody> </table>	Programming Notes		<p>This is one of the four RowStore Scratch Buffers which can be programmed to use the internal Media Cache (total size 640 CacheLine). When Intra Row Store Scratch Buffer Cache Select is programmed to "1", this data will be stored inside MFX Media Internal Stroage. Driver then needs to program this Base Address between 0 to 639, indicating starting cachelines address to Media Cache. Driver needs to make sure the whole buffer fits into MFX Media Internal Storage. (Notes: 1 cacheline per MB, and the buffer needs to have enough space for 1 MB row).</p>												
Programming Notes																	
<p>This is one of the four RowStore Scratch Buffers which can be programmed to use the internal Media Cache (total size 640 CacheLine). When Intra Row Store Scratch Buffer Cache Select is programmed to "1", this data will be stored inside MFX Media Internal Stroage. Driver then needs to program this Base Address between 0 to 639, indicating starting cachelines address to Media Cache. Driver needs to make sure the whole buffer fits into MFX Media Internal Storage. (Notes: 1 cacheline per MB, and the buffer needs to have enough space for 1 MB row).</p>																	
	5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
14	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
15:0	<p>Intra Row Store Scratch Buffer Base Address High</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Intra RowStore/Scratch Buffer Base Address This field is ignored in MPEG2 and VC1 mode.</p>	Format:	GraphicsAddress[47:32]														
Format:	GraphicsAddress[47:32]																
15	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	<p>Intra Row Store Scratch Buffer - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
0h	TRMODE_NONE	No tiled resource															
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
12	<p>Intra Row Store Scratch Buffer Cache Select This field controls if Intra Row Store is going to store inside Media Cache or to LLC.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Buffer going to LLC.</td> </tr> <tr> <td>1</td> <td></td> <td>Buffer going to Internal Media Storage</td> </tr> </tbody> </table>	Value	Name	Description	0		Buffer going to LLC.	1		Buffer going to Internal Media Storage							
Value	Name	Description															
0		Buffer going to LLC.															
1		Buffer going to Internal Media Storage															
11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
10	<p>Intra Row Store Scratch Buffer - Memory Compression Mode</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless</td> </tr> </tbody> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless														
Description																	
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless																	



MFX_PIPE_BUF_ADDR_STATE											
	<p>Memory Compression Enable is set to "1"</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode				
Value	Name										
0	Horizontal Compression Mode										
1	Vertical Compression Mode										
9	<p>Intra Row Store Scratch Buffer - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This surface is linear surface. This bit must be set to "0" since only TileY/TileYf/TileYs surface is allowed to be compressed</p>	Value	Name	0	Compression Disable						
Value	Name										
0	Compression Disable										
8:7	<p>Intra Row Store Scratch Buffer - Arbitration Priority Control</p> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
Value	Name										
00b	Highest priority										
01b	Second highest priority										
10b	Third highest priority										
11b	Lowest priority										
6:1	<p>Intra Row Store Scratch Buffer - Index to Memory Object Control State (MOCS) Tables</p> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.</p> <p>The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>										
0	Reserved										
16	<p>31:6 Deblocking Filter Row Store Scratch Base Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Deblocking Filter Row Store is needed for:</p> <ul style="list-style-type: none"> • AVC and VC1 In-Loop Deblocking Filter • VC1 Overlap-smoothing Filter • AVC, VC1, and MPEG-2 Out-Of-Loop Deblocking Filter (Intel extension) <p>This field provides the 64 byte aligned base address 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 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 Filter Row Store. Max 6 cachelines for VC1 and MPEG2, and max 4 for AVC (for MBAFF, 2 for non-MBAFF)</p> <p style="text-align: center;">Programming Notes</p>	Format:	GraphicsAddress[31:6]								
Format:	GraphicsAddress[31:6]										



MFX_PIPE_BUF_ADDR_STATE																
		<p>This is one of the four RowStore Scratch Buffers which can programmed to use the internal Media Cache (total size 640 Cacheline). When Deblocking Filter Row Store Scratch Buffer Cache Select is programmed to "1", this will be stored inside MFX Media Internal Storage. Driver then needs to program this Base Address between 0 to 639, indicating starting cachelines address location for this buffer. Driver needs to make sure the whole buffer fits into Media Internal Storage.</p> <p><i>(Notes: 2 cachelines per MB for non-mbaff; 4 cahcelines per MB pair for mbaff, and the buffer needs to have enough space for 1 MB (pair) row).</i></p>														
	5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ															
17	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
15:0	<p>Deblocking Filter Row Store Scratch Base Address High</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Deblocking Filter Row Store Scratch Buffer Address.</p>	Format:	GraphicsAddress[47:32]													
Format:	GraphicsAddress[47:32]															
18	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
	14:13	<p>Deblocking Filter Row Store - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved
Value	Name	Description														
0h	TRMODE_NONE	No tiled resource														
1h	TRMODE_TILEYF	4KB tiled resources														
2h	TRMODE_TILEYS	64KB tiled resources														
3h	Reserved															
12	<p>Deblocking Filter Row Store Scratch Buffer Cache Select This field controls if Intra Row Store is going to store inside Media Internal Storage or to LLC.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Buffer going to LLC</td> </tr> <tr> <td>1</td> <td></td> <td>Buffer going to Media Internal Storage</td> </tr> </tbody> </table>	Value	Name	Description	0		Buffer going to LLC	1		Buffer going to Media Internal Storage						
Value	Name	Description														
0		Buffer going to LLC														
1		Buffer going to Media Internal Storage														
	11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ															
	10	<p>Deblocking Filter Row Store Scratch - Memory Compression Mode</p> <table border="1"> <thead> <tr> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </tbody> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.												
Description																
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.																



MFX_PIPE_BUF_ADDR_STATE											
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode				
Value	Name										
0	Horizontal Compression Mode										
1	Vertical Compression Mode										
9	<p>Deblocking Filter Row Store Scratch - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This surface is linear surface. This bit must be set to "0" since only TileY/TileYf/TileYs surface is allowed to be compressed</p>	Value	Name	0	Compression Disable						
Value	Name										
0	Compression Disable										
8:7	<p>Deblocking Filter Row Store Scratch - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
Value	Name										
00b	Highest priority										
01b	Second highest priority										
10b	Third highest priority										
11b	Lowest priority										
6:1	<p>Deblocking Filter Row Store Scratch - Index to Memory Object Control State (MOCS) Tables</p> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.</p> <p>The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>										
0	Reserved										
19..50	<p>63:48 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>47:32 Reference Picture Address [n] High</p> <table border="1"> <tr> <td>Format:</td> <td>Address[47:32]</td> </tr> </table> <p>This field is for the upper range of Reference Picture Addresses</p> <p>31:6 Reference Picture Address [n]</p> <table border="1"> <tr> <td>Format:</td> <td>Address[31:6]</td> </tr> </table> <p>Specifies the 64 byte aligned reference frame buffer addresses for the motion compensation operation in AVC/ /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 L0 or L1 or both lists. But VC1 and MPEG2, worst case, can use up to 2 YUV frame-based surfaces for both forward and backward references:</p> <ul style="list-style-type: none"> • P-MB : RefAddr[0] - temporal closest previous field of a reference frame (can be the current frame) 	Format:	MBZ	Format:	Address[47:32]	Format:	Address[31:6]				
Format:	MBZ										
Format:	Address[47:32]										
Format:	Address[31:6]										



MFx_PIPE_BUF_ADDR_STATE																	
		<ul style="list-style-type: none"> RefAddr[1]- next temporal closest previous field of a reference frame (must be different from the current frame) <p>It is a variant (without the LongTermRefPic specification) of the RefFrameList[16] defined in AVC DXVA Spec. RefAddr[0-15] is indexed by frame_storeID »1. It is not a packed list, i.e. invalid entries can scatter among the list. All invalid addresses must be set to a valid address RefAddr[0] by the driver. The same applies to VC1 and MPEG2.</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <th colspan="2">Programming Notes</th> </tr> <tr> <td colspan="2">AVC: Always specifies all 16 addresses even some of them are not needed as indicated by the max num of active reference pictures. This is done for preventing data corruption (error, fault condition, etc.) by having all the references being set to a legal location.</td> </tr> </table>	Programming Notes		AVC: Always specifies all 16 addresses even some of them are not needed as indicated by the max num of active reference pictures. This is done for preventing data corruption (error, fault condition, etc.) by having all the references being set to a legal location.												
Programming Notes																	
AVC: Always specifies all 16 addresses even some of them are not needed as indicated by the max num of active reference pictures. This is done for preventing data corruption (error, fault condition, etc.) by having all the references being set to a legal location.																	
	5:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
51	31:15	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	Reference Picture - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode. <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
	0h	TRMODE_NONE	No tiled resource														
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
12:9	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
8:7	Reference Picture - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority						
Value	Name																
00b	Highest priority																
01b	Second highest priority																
10b	Third highest priority																
11b	Lowest priority																
6:1	Reference Picture - Index to Memory Object Control State (MOCS) Tables The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.																
	0	Reserved															



MFX_PIPE_BUF_ADDR_STATE																	
52	31:6	<p>Macroblock Buffer Base Address or Decoded Picture Error/Status Buffer Base Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>For decoder: Specifies the 64 byte aligned buffer address for writing a single error/status record into the memory when Pic Error/Status Report Enable is set in the MFX_PIPE_MODE_SELECT Command. The error/status record is written by HW at the end of decoding one single picture. The record is written in a fixed format, total 96-bits in size always. Please refer to "Media VDBOX -> Video Codec -> Other Codec Functions -> MFX Error Handling -> Decoder" session for the output format.</p> <p>For encoder: Specifies the 64 byte aligned buffer address for reading the per-MB indirect data from memory when MacroblockStatEnable is set in the MFX_AVC_IMG_STATE Command. 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.</p>	Format:	GraphicsAddress[31:6]													
	Format:	GraphicsAddress[31:6]															
5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
53	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
15:0	<p>Macroblock Buffer Base Address or Decoded Picture Error/Status Buffer Base Address High</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Macroblock Status Buffer Base Address</p>	Format:	GraphicsAddress[47:32]														
Format:	GraphicsAddress[47:32]																
54	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	<p>Macroblock Status Buffer - Tiled Resource Mode</p> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
0h	TRMODE_NONE	No tiled resource															
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
12:11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
10	<p>Macroblock Status Buffer - Memory Compression Mode</p> <table border="1"> <tr> <th style="text-align: center;">Description</th> </tr> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.														
Description																	
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.																	



MFX_PIPE_BUF_ADDR_STATE											
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode				
Value	Name										
0	Horizontal Compression Mode										
1	Vertical Compression Mode										
9	<p>Macroblock Status Buffer - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This surface is linear surface. This bit must be set to "0" since only TileY/TileYf/TileYs surface is allowed to be compressed</p>	Value	Name	0	Compression Disable						
Value	Name										
0	Compression Disable										
8:7	<p>Macroblock Status Buffer - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
Value	Name										
00b	Highest priority										
01b	Second highest priority										
10b	Third highest priority										
11b	Lowest priority										
6:1	<p>Macroblock Status Buffer - Index to Memory Object Control State (MOCS) Tables The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>										
0	Reserved										
55	<p>31:6 Macroblock ILDB StreamOut Buffer Base Address</p> <table border="1"> <tr> <td style="width: 20%;">Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Specifies the 64 byte aligned buffer address for writing MB ILDB parameter per MB to memory when Debocker streamout enable is set in the MFX_PIPE_MODE_SELECT Command. The ildb MB control parameters are written by HW at the end of each decoding MB. Only AVC edge information is being streamed out. It is used in AVC decode mode only.</p>	Format:	GraphicsAddress[31:6]								
Format:	GraphicsAddress[31:6]										
	<p>5:0 Reserved</p> <table border="1"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
56	<p>31:16 Reserved</p> <table border="1"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
	<p>15:0 Macroblock ILDB StreamOut Buffer Base Address High</p> <table border="1"> <tr> <td style="width: 20%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Deblocking Filter Row Store Scratch Address</p>	Format:	GraphicsAddress[47:32]								
Format:	GraphicsAddress[47:32]										



MFX_PIPE_BUF_ADDR_STATE																	
57	31:15	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	Macroblock ILDB StreamOut - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
	0h	TRMODE_NONE	No tiled resource														
	1h	TRMODE_TILEYF	4KB tiled resources														
	2h	TRMODE_TILEYS	64KB tiled resources														
	3h	Reserved															
	12:11	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
10	Macroblock ILDB StreamOut Buffer - Memory Compression Mode <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </tbody> </table> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 85%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td>1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Description		Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.		Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode						
Description																	
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.																	
Value	Name																
0	Horizontal Compression Mode																
1	Vertical Compression Mode																
9	Macroblock ILDB StreamOut Buffer - Memory Compression Enable <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Compression Disable</td> </tr> </tbody> </table> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">This surface is linear surface. This bit must be set to "0" since only TileY/TileYf/TileYs surface is allowed to be compressed</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	Programming Notes		This surface is linear surface. This bit must be set to "0" since only TileY/TileYf/TileYs surface is allowed to be compressed									
Value	Name																
0	Compression Disable																
Programming Notes																	
This surface is linear surface. This bit must be set to "0" since only TileY/TileYf/TileYs surface is allowed to be compressed																	
8:7	Macroblock ILDB StreamOut Buffer - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 85%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority						
Value	Name																
00b	Highest priority																
01b	Second highest priority																
10b	Third highest priority																
11b	Lowest priority																



MFX_PIPE_BUF_ADDR_STATE																	
	6:1	<p>Macroblock ILDB StreamOut Buffer - Index to Memory Object Control State (MOCS) Tables</p> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.</p> <p>The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>															
	0	Reserved															
58	31:6	<p>Second Macroblock ILDB StreamOut Buffer Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>64 byte aligned buffer.</p> <p>Specifies the 64 byte aligned buffer address for writing MB ILDB parameter per MB to memory when Debocker streamout enable is set in the MFX_PIPE_MODE_SELECT Command.</p> <p>The ildb MB control parameters are written by HW at the end of each decoding MB. Only AVC edge information is being streamed out. It is used in AVC decode mode only.</p>	Format:	GraphicsAddress[31:6]													
	Format:	GraphicsAddress[31:6]															
5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
59	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
15:0	<p>Second Macroblock ILDB StreamOut Buffer Base Address High</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Second Macroblock ILDB StreamOutBuffer Base Address.</p>	Format:	GraphicsAddress[47:32]														
Format:	GraphicsAddress[47:32]																
60	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	14:13	<p>Second Macroblock ILDB StreamOut Buffer - Tiled Resource Mode For Media Surfaces:</p> <p>This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Value	Name	Description														
0h	TRMODE_NONE	No tiled resource															
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
12:11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
Format:	MBZ																
10	<p>Second Macroblock ILDB StreamOut Buffer - Memory Compression Mode</p> <table border="1" style="width: 100%;"> <tr> <th style="text-align: center;">Description</th> </tr> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.														
Description																	
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.																	



MFX_PIPE_BUF_ADDR_STATE									
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode		
Value	Name								
0	Horizontal Compression Mode								
1	Vertical Compression Mode								
9	<p>Second Macroblock ILDB StreamOut Buffer - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This surface is linear surface. This bit must be set to "0" since only TileY/TileYf/TileYs surface is allowed to be compressed</p>	Value	Name	0	Compression Disable				
Value	Name								
0	Compression Disable								
8:7	<p>Second Macroblock ILDB StreamOut Buffer - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p>								
6:1	<p>Second Macroblock ILDB StreamOut Buffer - Index to Memory Object Control State (MOCS) Tables The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>								
0	Reserved								
61	<p>31 Reference Picture 15 - Memory Compression Mode</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Description									
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.									
Value	Name								
0	Horizontal Compression Mode								
1	Vertical Compression Mode								
	<p>30 Reference Picture 15 - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable		
Value	Name								
0	Compression Disable								
1	Compression Enable								
	<p>29 Reference Picture 14 - Memory Compression Mode</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </tbody> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.						
Description									
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.									



MFX_PIPE_BUF_ADDR_STATE									
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode		
Value	Name								
0	Horizontal Compression Mode								
1	Vertical Compression Mode								
28	<p>Reference Picture 14 - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable		
Value	Name								
0	Compression Disable								
1	Compression Enable								
27	<p>Reference Picture 13 - Memory Compression Mode</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Description									
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.									
Value	Name								
0	Horizontal Compression Mode								
1	Vertical Compression Mode								
26	<p>Reference Picture 13 - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable		
Value	Name								
0	Compression Disable								
1	Compression Enable								
25	<p>Reference Picture 12 - Memory Compression Mode</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Description									
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.									
Value	Name								
0	Horizontal Compression Mode								
1	Vertical Compression Mode								
24	<p>Reference Picture 12 - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable		
Value	Name								
0	Compression Disable								
1	Compression Enable								

**MFx_PIPE_BUF_ADDR_STATE**

23	Reference Picture 11 - Memory Compression Mode Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.						
	<table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>0</td><td>Horizontal Compression Mode</td></tr><tr><td>1</td><td>Vertical Compression Mode</td></tr></tbody></table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Value	Name						
0	Horizontal Compression Mode						
1	Vertical Compression Mode						
22	Reference Picture 11 - Memory Compression Enable						
	<table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>0</td><td>Compression Disable</td></tr><tr><td>1</td><td>Compression Enable</td></tr></tbody></table>	Value	Name	0	Compression Disable	1	Compression Enable
Value	Name						
0	Compression Disable						
1	Compression Enable						
21	Reference Picture 10 - Memory Compression Mode						
	<table border="1"><thead><tr><th colspan="2">Description</th></tr></thead><tbody><tr><td colspan="2">Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td></tr></tbody></table>	Description		Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.			
Description							
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.							
	<table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>0</td><td>Horizontal Compression Mode</td></tr><tr><td>1</td><td>Vertical Compression Mode</td></tr></tbody></table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Value	Name						
0	Horizontal Compression Mode						
1	Vertical Compression Mode						
20	Reference Picture 10 - Memory Compression Enable						
	<table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>0</td><td>Compression Disable</td></tr><tr><td>1</td><td>Compression Enable</td></tr></tbody></table>	Value	Name	0	Compression Disable	1	Compression Enable
Value	Name						
0	Compression Disable						
1	Compression Enable						
19	Reference Picture 9 - Memory Compression Mode						
	<table border="1"><thead><tr><th colspan="2">Description</th></tr></thead><tbody><tr><td colspan="2">Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td></tr></tbody></table>	Description		Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.			
Description							
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.							
	<table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>0</td><td>Horizontal Compression Mode</td></tr><tr><td>1</td><td>Vertical Compression Mode</td></tr></tbody></table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Value	Name						
0	Horizontal Compression Mode						
1	Vertical Compression Mode						



MFX_PIPE_BUF_ADDR_STATE							
18	Reference Picture 9 - Memory Compression Enable						
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable
	Value	Name					
0	Compression Disable						
1	Compression Enable						
17	Reference Picture 8 - Memory Compression Mode						
	Description						
	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.						
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Value	Name						
0	Horizontal Compression Mode						
1	Vertical Compression Mode						
16	Reference Picture 8 - Memory Compression Enable						
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable
	Value	Name					
0	Compression Disable						
1	Compression Enable						
15	Reference Picture 7 - Memory Compression Mode						
	Description						
	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.						
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Value	Name						
0	Horizontal Compression Mode						
1	Vertical Compression Mode						
14	Reference Picture 7 - Memory Compression Enable						
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable
	Value	Name					
0	Compression Disable						
1	Compression Enable						
13	Reference Picture 6 - Memory Compression Mode						
	Description						
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.							



MFX_PIPE_BUF_ADDR_STATE									
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode		
Value	Name								
0	Horizontal Compression Mode								
1	Vertical Compression Mode								
12	<p>Reference Picture 6 - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable		
Value	Name								
0	Compression Disable								
1	Compression Enable								
11	<p>Reference Picture 5 - Memory Compression Mode</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Description									
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.									
Value	Name								
0	Horizontal Compression Mode								
1	Vertical Compression Mode								
10	<p>Reference Picture 5 - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable		
Value	Name								
0	Compression Disable								
1	Compression Enable								
9	<p>Reference Picture 4 - Memory Compression Mode</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Description	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
Description									
Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.									
Value	Name								
0	Horizontal Compression Mode								
1	Vertical Compression Mode								
8	<p>Reference Picture 4 - Memory Compression Enable</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Compression Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable		
Value	Name								
0	Compression Disable								
1	Compression Enable								



MFX_PIPE_BUF_ADDR_STATE

7	Reference Picture 3 - Memory Compression Mode	
	Description	
	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.	
	Value	Name
	0	Horizontal Compression Mode
1	Vertical Compression Mode	
6	Reference Picture 3 - Memory Compression Enable	
	Value	Name
	0	Compression Disable
1	Compression Enable	
5	Reference Picture 2 - Memory Compression Mode	
	Description	
	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.	
	Value	Name
	0	Horizontal Compression Mode
1	Vertical Compression Mode	
4	Reference Picture 2 - Memory Compression Enable	
	Value	Name
	0	Compression Disable
1	Compression Enable	
3	Reference Picture 1 - Memory Compression Mode	
	Description	
	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.	
	Value	Name
	0	Horizontal Compression Mode
1	Vertical Compression Mode	



MFX_PIPE_BUF_ADDR_STATE												
	2	Reference Picture 1 - Memory Compression Enable <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Compression Disable</td> </tr> <tr> <td>1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable				
	Value	Name										
	0	Compression Disable										
	1	Compression Enable										
	1	Reference Picture 0 - Memory Compression Mode <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.</td> </tr> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> <tr> <td>0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td>1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Description		Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.		Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode
	Description											
	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details. Note: This bit is not used unless Memory Compression Enable is set to "1" Note: This is a READ Surface. The setting of this bit should match the settings on how this is written out before.											
	Value	Name										
	0	Horizontal Compression Mode										
	1	Vertical Compression Mode										
0	Reference Picture 0 - Memory Compression Enable <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Compression Disable</td> </tr> <tr> <td>1</td> <td>Compression Enable</td> </tr> </tbody> </table>	Value	Name	0	Compression Disable	1	Compression Enable					
Value	Name											
0	Compression Disable											
1	Compression Enable											
62	31:6 Scaled Reference Surface Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Specifies the 64 byte aligned down scaled reference frame buffer addresses that needs to be used by the PAK down-scaler to write the down scaled pixels. Only the luma pixels will be downscaled and written to the surface</p>	Format:	GraphicsAddress[31:6]									
Format:	GraphicsAddress[31:6]											
	5:0 Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
63	31:16 Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
	15:0 Scaled Reference Surface Base Address High <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>This field is for the upper range of Scaled Reference Surface Base Address.</p>	Format:	GraphicsAddress[47:32]									
Format:	GraphicsAddress[47:32]											
64	31:15 Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
	14:13 Scaled Reference Surface - Tiled Resource Mode For Media Surfaces: This field specifies the tiled resource mode <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource					
Value	Name	Description										
0h	TRMODE_NONE	No tiled resource										



MFX_PIPE_BUF_ADDR_STATE		
	1h	TRMODE_TILEYF No tiled resource
	2h	TRMODE_TILEYS No tiled resource
	3h	Reserved
12:11	Reserved	
	Format:	MBZ
10	Scaled Reference Surface - Memory Compression Mode	
	Description	
	Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter -section Media Memory Compression for more details.	
	Value	Name
	0	Horizontal Compression Mode
	1	Vertical Compression Mode
9	Scaled Reference Surface - Memory Compression Enable	
	Format:	Enable
	Memory compression shouldnt be enabled for this surface.	
8:7	Scale Reference Surface - 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 Priority
	11b	Lowest Priority
6:1	Scaled Reference Surface - Index to Memory Object Control State (MOCS) Tables	
	The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.	
	The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
0	Reserved	



MFX_PIPE_MODE_SELECT

MFX_PIPE_MODE_SELECT			
Source:	VideoCS		
Length Bias:	2		
<p>Specifies which codec and hardware module is being used to encode/decode the video data, on a per-frame basis.</p> <p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_COMMON
		Format:	OpCode
	26:24	Opcode	
		Default Value:	0h MFX_COMMON_STATE
		Format:	OpCode
	23:21	SubOpA	
Default Value:		0h	
Format:		OpCode	
20:16	SubOpB		
	Default Value:	0h MFX_PIPE_MODE_SELECT	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length	Format:	=n Total Length - 2
	Value	Name	Description
	3h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)
1	31	Reserved	
	30	Reserved	
		Format:	MBZ



MFX_PIPE_MODE_SELECT		
29	Reserved	
28:27	Reserved	
26	Reserved	
	Format:	MBZ
25	Reserved	
	Format:	MBZ
24	Reserved	
	Format:	MBZ
23:19	Reserved	
	Format:	MBZ
18	Reserved	
17	Decoder Short Format Mode For IT mode, this bit must be 0.	
	Value	Description
	0	Short Format Driver Interface [Default] AVC/VC1/MVC/VP8 Short Format Mode is in use Note: There is no Short Format for VP8 yet, so this field must be set to 1 for VP8.
	1	Long Format Driver Interface AVC/VC1/MVC/VP8 Long Format Mode is in use.
16:15	Decoder Mode select Each coding standard supports two entry points: VLD entry point and IT (IDCT) entry point. This field selects which one is in use. This field is only valid if Codec Select is 0 (decoder).	
	Value	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 Engine for IT Mode Note: Only VC1 and MPEG2 support this mode
	2h	Deblocker Mode Configure the MFD Engine for Standalone Deblocker Mode. Require streamout AVC edge control information from preceeding decoding pass.
	3h	Reserved
14	Reserved	
13	Reserved	
12	Deblocker Stream-Out Enable This field indicates if Deblocker information is going to be streamout during VLD decoding. For AVC, it is needed to enable the deblocker streamout as the AVC Disable_DLKFilterIdc is a slice level parameters. Driver needs to determine ahead of time if at least one slice of the current frame/ has deblocker ON.	



MFX_PIPE_MODE_SELECT									
	Value	Name	Description						
	0h	Disable							
	1h	Enable							
11	<p>Pic Error/Status Report Enable.</p> <p>This field control whether the error/status reporting is enable or not.0: Disable1: EnableIn decoder modes: Error reporting is written out once per frame. The Error Report frame ID listed in DW3 along with the VLD/IT error status bits are packed into one cache and written to the "Decoded Picture Error/Status Buffer address" listed in the MFX_PIPE_BUF_ADDR_STATE Command. Note: driver shall program different error buffer addresses between pictures; otherwise, hardware might overwrite previous written data if driver does not read it fast enough.In encoder modes: Not used</p> <p>Please refer to "Media VDBOX -> Video Codec -> Other Codec Functions -> MFX Error Handling -> Decoder" session for the output format.</p> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> </tr> <tr> <td>1h</td> <td>Enable</td> </tr> </tbody> </table>			Value	Name	0h	Disable	1h	Enable
Value	Name								
0h	Disable								
1h	Enable								
10	<p>Stream-Out Enable</p> <p>This field controls whether the macroblock parameter stream-out is enabled during VLD decoding for transcoding purpose.</p> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> </tr> <tr> <td>1h</td> <td>Enable</td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 10px;">Programming Notes</p> <p>In decoder modes: The Stream-Out feature is added to support transcoding. While decoding 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.</p>			Value	Name	0h	Disable	1h	Enable
Value	Name								
0h	Disable								
1h	Enable								
9	<p>Post Deblocking Output Enable (PostDeblockOutEnable)</p> <p>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.</p> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> </tr> <tr> <td>1h</td> <td>Enable</td> </tr> </tbody> </table>			Value	Name	0h	Disable	1h	Enable
Value	Name								
0h	Disable								
1h	Enable								



MFX_PIPE_MODE_SELECT			
8	Pre Deblocking Output Enable (PreDeblockOutEnable) This field controls the output write for the reconstructed pixels BEFORE the deblocking filter.		
	Value	Name	
	0h	Disable	
	1h	Enable	
7	Scaled Surface Enable This field indicates if the scaled surface is enabled. This field enables the 4x HME downscaler of the reconstructed image. Only supported for AVC and VP8 formats.		
	Value	Name	
	0h	Disable	
	1h	Enable	
6	Frame Statistics StreamOut Enable This field controls the frame level statistics streamout from the PAK.		
	Value	Name	
	0h	Disable	
	1h	Enable	
5	Stitch Mode Exists If: //CodecSel=Encode and StandardSel=AVC		
	Value	Name	
	Value	Name	Description
	0h	Not in stitch mode	
	1h	In the special stitch mode	This mode can be used for any Codec as long as bitfield conditions are met.
4	Codec Select		
	Value	Name	Description
	0h	Decode	
	1h	Encode	Valid only if StandardSel is AVC and MPEG2)
3:0	Standard Select		
	Value	Name	Description
	0000b	MPEG2	
	0001b	VC1	
	0010b	AVC	Covers both AVC and MVC
	0011b	JPEG	
	0101b	VP8	Decoder, Encoder
	0110b	Reserved	
	0111b	Reserved	
	1111b	UVLD	SW decoder w/ embedded micro-controller and co-processor



MFX_PIPE_MODE_SELECT										
2	31	Reserved								
	Format: MBZ									
	30	Reserved								
	29	Reserved								
	Format: MBZ									
	28	Reserved								
	27	Reserved								
	26	Reserved								
	25	Reserved								
	24	Reserved								
	Format: MBZ									
	23	Reserved								
	22:21	Reserved								
	20:19	Reserved								
	Format: MBZ									
	18	Reserved								
	Format: MBZ									
	17	Reserved								
	16	Reserved								
	15	Reserved								
14	VLF 720i (Odd Height) in VC1 Mode This bit indicates VLF write out VC1 picture with odd height (in MBs).									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Disable [Default]</td> <td></td> </tr> <tr> <td style="text-align: center;">1</td> <td>Enable</td> <td>720i Enable</td> </tr> </tbody> </table>		Value	Name	Description	0	Disable [Default]		1	Enable	720i Enable
Value	Name	Description								
0	Disable [Default]									
1	Enable	720i Enable								
13	Reserved									
Format: MBZ										
12	Reserved									
11	Reserved									
10	MPC pref08x8_disable Flag (Default 0)									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Disable</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Enable</td> </tr> </tbody> </table>		Value	Name	0	Disable	1	Enable			
Value	Name									
0	Disable									
1	Enable									
9	Reserved									
Format: MBZ										



MFX_PIPE_MODE_SELECT											
8	Reserved										
7	Reserved										
6	Clock gate Enable at Slice-level BitFieldDesc: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable</td> <td>Disable Slice-level Clock gating, Unit-level Clock gating will apply</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Enable Slice-level Clock gating, overrides any Unit level Clock gating</td> </tr> </tbody> </table>		Value	Name	Description	0	Disable	Disable Slice-level Clock gating, Unit-level Clock gating will apply	1	Enable	Enable Slice-level Clock gating, overrides any Unit level Clock gating
Value	Name	Description									
0	Disable	Disable Slice-level Clock gating, Unit-level Clock gating will apply									
1	Enable	Enable Slice-level Clock gating, overrides any Unit level Clock gating									
5	Reserved										
4	Reserved Format: MBZ										
3	VDS ILDB Calculation This bit forces all MB into INTRA MBs before doing ILDB control generation in VDS. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable [Default]</td> <td>Use original definition for ILDB calculation.</td> </tr> <tr> <td>1</td> <td>Enable</td> <td>Force neighbor Intra MB = 1 on ILDB BS calculation.</td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p style="text-align: center;">Programming Notes</p> <p>When the bit is '0', the ILDB control generation will be the same as the original spec (AVC/VC1).</p> </div>		Value	Name	Description	0	Disable [Default]	Use original definition for ILDB calculation.	1	Enable	Force neighbor Intra MB = 1 on ILDB BS calculation.
Value	Name	Description									
0	Disable [Default]	Use original definition for ILDB calculation.									
1	Enable	Force neighbor Intra MB = 1 on ILDB BS calculation.									
2:1	Reserved										
0	Reserved										
3	31:0	Pic Status/Error Report ID Exists If: //Decoder Mode Only Format: U32 In decoder modes: Error reporting is written out once per frame. This field along with the VLD error status bits are packed into one cache and written to the memory location specified by "Decoded Picture Error/Status Buffer address" listed in the MFX_PIPE_BUF_ADDR_STATE Command. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>32-bit unsigned</td> <td>Unique ID Number</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	32-bit unsigned	Unique ID Number	1h	Reserved	
Value	Name	Description									
0h	32-bit unsigned	Unique ID Number									
1h	Reserved										
4	31:0	Reserved Format: MBZ									



MFX_QM_STATE

MFX_QM_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_MULTI_DW
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	0h MFX_COMMON_STATE
		Format:	OpCode
	23:21	SubOpcode A	
Default Value:		0h	
Format:		OpCode	
20:16	SubOpcode B		
	Default Value:	7h	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	20h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:2	Reserved	
		Format:	MBZ
1	1:0	AVC	
		Exists If:	//AVC- Decoder Only



MFX_QM_STATE											
	<p>For AVC QM Type: This field specifies which Quantizer Matrix is loaded.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>AVC_4x4_Intra_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)</td> </tr> <tr> <td style="text-align: center;">1</td> <td>AVC_4x4_Inter_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)</td> </tr> <tr> <td style="text-align: center;">2</td> <td>AVC_8x8_Intra_MATRIX</td> </tr> <tr> <td style="text-align: center;">3</td> <td>AVC_8x8_Inter_MATRIX</td> </tr> </tbody> </table>	Value	Name	0	AVC_4x4_Intra_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)	1	AVC_4x4_Inter_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)	2	AVC_8x8_Intra_MATRIX	3	AVC_8x8_Inter_MATRIX
Value	Name										
0	AVC_4x4_Intra_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)										
1	AVC_4x4_Inter_MATRIX, (Y-4DWs, Cb-4DWs, Cr-4DWs, reserved-4DWs)										
2	AVC_8x8_Intra_MATRIX										
3	AVC_8x8_Inter_MATRIX										
	<p>1:0 MPEG2</p> <p>Exists If: //MPEG2- Decoder Only</p> <p>For MPEG2 QM Type: This field specifies which Quantizer Matrix is loaded.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>MPEG_INTRA_QUANTIZER_MATRIX</td> </tr> <tr> <td style="text-align: center;">1</td> <td>MPEG_NON_INTRA_QUANTIZER_MATRIX</td> </tr> <tr> <td style="text-align: center;">2-3</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	0	MPEG_INTRA_QUANTIZER_MATRIX	1	MPEG_NON_INTRA_QUANTIZER_MATRIX	2-3	Reserved		
Value	Name										
0	MPEG_INTRA_QUANTIZER_MATRIX										
1	MPEG_NON_INTRA_QUANTIZER_MATRIX										
2-3	Reserved										
	<p>1:0 JPEG</p> <p>Exists If: //JPEG- Encoder Only</p> <p>For JPEG QM Type: This field specifies which Quantizer Matrix is loaded.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>JPEG_Luma_Y_QUANTIZER_MATRIX (or R)</td> </tr> <tr> <td style="text-align: center;">1</td> <td>JPEG_Chroma_Cb_QUANTIZER_MATRIX (or G)</td> </tr> <tr> <td style="text-align: center;">2</td> <td>JPEG_Chroma_Cr_QUANTIZER_MATRIX (or B)</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>For JPEG encoder, each quantization element presents 16-bit 1/QM[i][j]. In RGB encoding, because the order input image components can be RGB, GBR, BGR, YUV, the value 0 is used for the first image component, the value 1 is used for the second image component, and the value 2 is used for the third image component.</p>	Value	Name	0	JPEG_Luma_Y_QUANTIZER_MATRIX (or R)	1	JPEG_Chroma_Cb_QUANTIZER_MATRIX (or G)	2	JPEG_Chroma_Cr_QUANTIZER_MATRIX (or B)		
Value	Name										
0	JPEG_Luma_Y_QUANTIZER_MATRIX (or R)										
1	JPEG_Chroma_Cb_QUANTIZER_MATRIX (or G)										
2	JPEG_Chroma_Cr_QUANTIZER_MATRIX (or B)										
2..33	<p>31:0 Forward Quantizer Matrix</p> <p>Format: U32</p> <p>The format of a Quantizer Matrix is an 8x8 matrix in raster order. Each element is an unsigned byte.</p>										



MFX_STATE_POINTER

MFX_STATE_POINTER			
Source:	VideoCS		
Length Bias:	2		
<p>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.</p> <p>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.</p> <p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFX_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	0h MFX_COMMON_STATE
		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	



MFX_STATE_POINTER																	
	11:0	DWord Length															
		<table border="1"> <tr> <td>Default Value:</td> <td>0h DWORD_COUNT_n</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table>	Default Value:	0h DWORD_COUNT_n	Format:	=n Total Length - 2											
Default Value:	0h DWORD_COUNT_n																
Format:	=n Total Length - 2																
1	31:5	State Pointer															
		<table border="1"> <tr> <td>Format:</td> <td>GeneralStateOffset[31:5]Indirect State Buffer</td> </tr> </table> <p>Specifies the 32-byte aligned address of an Indirect State Buffer. This pointer is relative to the General State Base Address.</p>	Format:	GeneralStateOffset[31:5]Indirect State Buffer													
	Format:	GeneralStateOffset[31:5]Indirect State Buffer															
4:2	Reserved																
		<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ																
	1:0	State Pointer Index															
		Specifies one of the four indirect state pointers to program.															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td></td> <td>indirect state pointer 0 (image state)</td> </tr> <tr> <td>01b</td> <td></td> <td>indirect state pointer 1 (slice state)sc</td> </tr> <tr> <td>10b</td> <td></td> <td>indirect state pointer 2</td> </tr> <tr> <td>11b</td> <td></td> <td>indirect state pointer 3</td> </tr> </tbody> </table>	Value	Name	Description	00b		indirect state pointer 0 (image state)	01b		indirect state pointer 1 (slice state)sc	10b		indirect state pointer 2	11b		indirect state pointer 3
Value	Name	Description															
00b		indirect state pointer 0 (image state)															
01b		indirect state pointer 1 (slice state)sc															
10b		indirect state pointer 2															
11b		indirect state pointer 3															



MFX_STITCH_OBJECT

MFX_STITCH_OBJECT		
Source:	VideoCS	
Length Bias:	2	
<p>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.</p> <p>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. Context switch interrupt is not supported by this command.</p> <p>The input bitstreams may be elementary streams stored in protected memory.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
	Format: OpCode	
	28:27	Pipeline
		Default Value: 2h MFC_STITCH_OBJECT
	Format: OpCode	
	26:24	Media Command Opcode
		Default Value: 0h MFX_COMMON
	Format: OpCode	
	23:21	SubOpcode A
		Default Value: 2h
	Format: OpCode	
	20:16	SubOpcode B
		Default Value: Ah
Format: OpCode		
15:12	Reserved	
	Format: MBZ	
11:0	DWord Length	
	Default Value: 0h Excludes DWord (0,1) = Variable Length in DW (>= 3)	
	Format: =n Total Length - 2	
If it is 3, it indicates the absent of inline data.		



MFX_STITCH_OBJECT						
1	31:18	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	17:16	<p>Source Data Starting Byte Offset Source Data Starting Byte Position within the very first inline DW.</p>				
	15:14	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	13:8	<p>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.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[1,32]</td> <td></td> </tr> </tbody> </table>	Value	Name	[1,32]	
	Value	Name				
	[1,32]					
	7:4	Reserved				
	3	Reserved				
2	<p>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, hardware ignores this bit.</p>					
1	<p>Last Destination Data Insert Command Flag</p> <table border="1" style="width: 100%;"> <tr> <td>THIS FIELD MUST BE THE SAME AS Last Source Header Data Insert Command Flag</td> </tr> <tr> <td>No more insertion command and no more PAK-OBJECT command follows. Flush data out to memory</td> </tr> </table>	THIS FIELD MUST BE THE SAME AS Last Source Header Data Insert Command Flag	No more insertion command and no more PAK-OBJECT command follows. Flush data out to memory			
THIS FIELD MUST BE THE SAME AS Last Source Header Data Insert Command Flag						
No more insertion command and no more PAK-OBJECT command follows. Flush data out to memory						
0	Reserved					
2	31:19	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
18:0	<p>Indirect Data Length</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U19</td> </tr> </table> <p>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.</p>	Format:	U19			
Format:	U19					
3	31:0	<p>Indirect Data Start Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MfxIndirectBitstreamObjectAddress[31:0]</td> </tr> </table> <p>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 MFX Indirect Bitstream Object Base Address. Hardware ignores this field if indirect data is not present.</p>	Format:	MfxIndirectBitstreamObjectAddress[31:0]		
Format:	MfxIndirectBitstreamObjectAddress[31:0]					
4..n	31:0	<p>Insert Data Payload Inline data to be inserted to the output bitstream buffer</p>				



MFX_SURFACE_STATE

MFX_SURFACE_STATE	
Source:	VideoCS
Length Bias:	2
<p>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:</p> <ul style="list-style-type: none">• Uncompressed, original input picture to be encoded• Reconstructed non-filtered/filtered display picture (becoming reference pictures as well for subsequent temporal inter-prediction) <p>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.</p> <p>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 Gen7 MFX :</p> <ul style="list-style-type: none">• 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 <p>Note that the following data structures are not specified through the media surface state</p> <ul style="list-style-type: none">• 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). <p>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.</p> <p>For non pixel data, such as row stores, indirect data (Compressed Slice Data, AVC MV record, Coeff record and</p>	



MFX_SURFACE_STATE

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 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 intel decoder pipelineand 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 constrained 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 Type	
		Default Value: 3h PARALLEL_VIDEO_PIPE	
		Format: OpCode	
	28:27	Pipeline	Default Value: 2h MFX_COMMON
			Format: OpCode
			Opcode
	26:24	Opcode	Default Value: 0h MFX_COMMON_STATE
			Format: OpCode
			SubOpA
	23:21	SubOpA	Default Value: 0h
			Format: OpCode
			SubOpB
20:16	SubOpB	Default Value: 1h	
		Format: OpCode	
		Reserved	
15:12	Reserved	Format: MBZ	



MFX_SURFACE_STATE																							
	11:0	DWord Length																					
		Format: =n Total Length - 2																					
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>4h</td> <td>DWORD_COUNT_n [Default]</td> <td>Excludes DWord (0,1)</td> </tr> </tbody> </table>	Value	Name	Description	4h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)															
Value	Name	Description																					
4h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)																					
1	31:4	Reserved																					
		Format: MBZ																					
	3:0	Surface Id																					
		Format: U4																					
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Reserved</td> <td></td> </tr> <tr> <td>0001b</td> <td>Reserved</td> <td></td> </tr> <tr> <td>0010b</td> <td>Reserved</td> <td></td> </tr> <tr> <td>0011b</td> <td>Reserved</td> <td></td> </tr> <tr> <td>0100b</td> <td>Source Input Picture (encoder)</td> <td>8-bit uncompressed data</td> </tr> <tr> <td>0101b</td> <td>Reconstructed Scaled Reference Picture</td> <td>8-bit data</td> </tr> </tbody> </table>	Value	Name	Description	0000b	Reserved		0001b	Reserved		0010b	Reserved		0011b	Reserved		0100b	Source Input Picture (encoder)	8-bit uncompressed data	0101b	Reconstructed Scaled Reference Picture	8-bit data
		Value	Name	Description																			
		0000b	Reserved																				
0001b	Reserved																						
0010b	Reserved																						
0011b	Reserved																						
0100b	Source Input Picture (encoder)	8-bit uncompressed data																					
0101b	Reconstructed Scaled Reference Picture	8-bit data																					
2	31:18	Height																					
		Format: U14-1 Height																					
		<p>This field specifies the height of the Picture in units of pixels/residuals. For PLANAR surface formats, this field indicates the height of the Y (luma) plane. Note : Gen7 Video Codecs must program less than and equal to 4K.(In future, it will be ideal to have this field define in a WORD boundary.)AVC - multiple of 2 MB rows for field pictureVC1 - multiple of 4 pixels for field pictureMPEG2 - multiple of 2 MB rows for field picJPEG - multiple of integral MCU (8 or 16 pixels) per picture</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> <td>representing heights [1,16384]</td> </tr> </tbody> </table>	Value	Name	Description	[0,16383]		representing heights [1,16384]															
		Value	Name	Description																			
[0,16383]		representing heights [1,16384]																					
<table border="1"> <thead> <tr> <th colspan="3" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="3"> <ul style="list-style-type: none"> 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. </td> </tr> </tbody> </table>	Programming Notes			<ul style="list-style-type: none"> 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. 																			
Programming Notes																							
<ul style="list-style-type: none"> 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. 																							



MFX_SURFACE_STATE							
17:4	Width						
	Format:	U14-1 Width					
	<p>This field specifies the width of the Picture in units of pixels/residuals. For PLANAR surface formats, this field indicates the width of the Y (luma) plane.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> <td>representing widths [1,16384]</td> </tr> </tbody> </table>		Value	Name	Description	[0,16383]	
Value	Name	Description					
[0,16383]		representing widths [1,16384]					
Programming Notes							
<ul style="list-style-type: none"> 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). Width (field value + 1) must be a multiple of 2 for PLANAR_420, MFX HW does not use this field, the picture width is read from IMG State instead, because this field may not equal to the actual picture width. This field is used by the KMD to allocate surface in GTT. 							
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					
	<p>Specifies the distance to the U/V values with respect to the even numbered Y channels in the V direction</p>						
Programming Notes							
This field is ignored for all formats except PLANAR_420_8							
3	31:28	Surface Format					
		Format:	U4				
		<p>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.</p>					
		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					



MFX_SURFACE_STATE				
	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	
27	Interleave Chroma			
	Format:		Enable	
	This field indicates that the chroma fields are interleaved in a single plane rather than stored as two separate planes. This field is only used for PLANAR surface formats. For AVC/VC1/MPEG VLD and IT modes : set to Enable to support interleave U/V only. For JPEG : set to Disable for all formats (including 4:2:0) - because JPEG does not support NV12. (This field is needed only if JPEG will support NV12; otherwise is ignored.)			
	Value	Name		
	1	Enable		
	0	Disable		
26:20	Reserved			
	Format:		MBZ	
19:3	Surface Pitch			
	Format:		U17-1 pitch in Bytes	
	This field specifies the surface pitch in (#Bytes).			
	Value	Name	Description	
	[0,2047]		to [1B, 2048B]	
	Programming Notes			
	For tiled surfaces, the pitch must be a multiple of the tile width (i.e.128 bytes aligned). If Half Pitch for Chroma is set, this field must be a multiple of two tile widths for tiled surfaces, or a multiple of 2 bytes for linear surfaces. For Y-tiled surfaces: Range = [127, 524287] to [128B,256KB] = [1 tile, 2048 tiles]			
	If Media Memory Compression is enabled, the following max pitch size restriction must be honored. For larger resolution, Media Memory compression Must be disabled.			
	Tiling Mode	Pixel Format	Max Frame Width (bytes)	Max Frame Width (pixels)
	Legacy 4K	8bpp	16k	16k
		16bpp	16k	8k
		32bpp	16k	4k
		64bpp	16k	2k
		128bpp	16k	1k
				Max Pitch (bytes)
				16k + 127
				16k + 127
				16k + 127
				16k + 127
				16k + 127



MFX_SURFACE_STATE

MFX_SURFACE_STATE						
		TileYF	8bpp	8k	8k	8k + 63
			16bpp	16k	8k	16k + 127
			32bpp	16k	4k	16k + 127
			64bpp	16k	2k	16k + 255
			128bpp	16k	1k	16k + 255
		TileYS	8bpp	16k	16k	16k + 255
			16bpp	16k	8k	16k + 511
			32bpp	16k	4k	16k + 511
			64bpp	16k	2k	16k + 1023
			128bpp	16k	1k	16k + 1023
2	Half Pitch for Chroma					
	Format:		Enable			
	(This field must be set to Disable)This field indicates that the chroma plane(s) will use a pitch equal to half the value specified in the Surface Pitch field. This field is only used for PLANAR surface formats.This field is ignored by MFX (unless we support YV12)					
1	Tiled Surface					
	Format:		Boolean			
	(This field must be set to TRUE: Tiled)This field specifies whether the surface is tiled.This field is ignored by MFX					
	Value	Name	Description			
	0	False	Linear			
	1	True	Tiled			
	Programming Notes					
	Linear surfaces can be mapped to Main Memory (uncached) or System Memory (cacheable, snooped). Tiled surfaces can only be mapped to Main Memory.The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state 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_XMAJOR			
	1h	YMAJOR	TILEWALK_YMAJOR			



MFX_SURFACE_STATE				
		<p style="text-align: center;">Programming Notes</p> <p>The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this bit</p>		
4	31	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	30:16	<p>X Offset for U(Cb)</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">U15 Pixel Offset</td> </tr> </table> <p>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)</p> <p style="text-align: center;">Programming Notes</p> <p>For PLANAR_420 and PLANAR_422 surface formats, this field must be zero.</p>	Format:	U15 Pixel Offset
	Format:	U15 Pixel Offset		
	15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
14:0	<p>Y Offset for U(Cb)</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">U15 Pixel Row Offset</td> </tr> </table> <p>This field specifies the vertical offset in rows 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.</p> <p style="text-align: center;">Programming Notes</p> <p>For PLANAR_420 and PLANAR_422 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.</p>	Format:	U15 Pixel Row Offset	
Format:	U15 Pixel Row Offset			
31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
5	28:16	<p>X Offset for V(Cr)</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">U13 Offset in Pixels</td> </tr> </table> <p>This field must be zero for NV12 and IMC 1 and 3</p> <p>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.</p> <p style="text-align: center;">Programming Notes</p> <p>For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.</p>	Format:	U13 Offset in Pixels
	Format:	U13 Offset in Pixels		



MFX_SURFACE_STATE				
15:0	<p>Y Offset for V(Cr)</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U16 Row Offset in Pixels</td> </tr> </table> <p>This field specifies the vertical 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.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>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.</p>	Format:	U16 Row Offset in Pixels	Programming Notes
Format:	U16 Row Offset in Pixels			
Programming Notes				



MFX_VC1_DIRECTMODE_STATE

MFX_VC1_DIRECTMODE_STATE			
Source:	VideoCS		
Length Bias:	2		
Exists If:	//VC1 decoding in VLD modes		
<p>This is a picture level command and should be issued only once, even for a multi-slices picture. 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 bits per MB, to store the top and bottom field MVs (32-bit MV_{x,y} each).</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_VC1_DIRECTMODE_STATE
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	2h VC1_COMMON
		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:	0005h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:6	<p>Direct MV Write Buffer Base Address for the Current Picture</p> <p>This field provides the base address of the DMV write buffer to store the motion vectors decoded 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 the hardware assumes frame width (in MBs) fixed at 128 (smallest power of 2 value larger than 120 - 1920x1088 screen resolution). This field is only valid for a P picture</p>	



MFX_VC1_DIRECTMODE_STATE																	
	5:0	Reserved															
		Format: MBZ															
2	31:16	Reserved															
		Format: MBZ															
	15:0	Direct MV Write Buffer Base Address for the Current Picture [47:32] This field is for the upper range of Direct MV Write Buffer Base Address for the Current Picture. This field is used for 48-bit addressing.															
3	31:15	Reserved															
		Format: MBZ															
	14:13	Direct MV Write Buffer - Tiled Resource Mode															
		Format: U2															
		For Media Surfaces: This field specifies the tiled resource mode.															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
Value	Name	Description															
0h	TRMODE_NONE	No tiled resource															
1h	TRMODE_TILEYF	4KB tiled resources															
2h	TRMODE_TILEYS	64KB tiled resources															
3h	Reserved																
	12:11	Reserved															
		Format: MBZ															
	10	Direct MV Write Buffer - Memory Compression Mode															
		Format: U1															
		Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter, Media Memory Compression section for more details.															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td>1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode									
Value	Name																
0	Horizontal Compression Mode																
1	Vertical Compression Mode																
	9	Direct MV Write Buffer - Memory Compression Enable															
		Format: Enable															
		Memory compression will be attempted for this surface.															
	8:7	Direct MV Write Buffer Base Address for the Current Picture - Arbitration Priority Control															
		Format: U2															
		This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority									
Value	Name																
00b	Highest priority																
01b	Second highest priority																



MFV_VC1_DIRECTMODE_STATE		
	10b	Third highest priority
	11b	Lowest priority
6:1	Direct MV Write Buffer - Index to Memory Object Control State (MOCS) Tables	
	Format:	U6
	The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
	0	Reserved
4	31:6	Direct MV Read Buffer Base Address for the Reference Picture This field provides 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 cacheline aligned. This field is only valid for a B picture.
	5:0	Reserved
	Format:	MBZ
5	31:16	Reserved
	Format:	MBZ
	Reserved for 64-bit address extension.	
	15:0	Direct MV Read Buffer Base Address for the Current Picture [47:32] This field is for the upper range of Direct MV Read Buffer Base Address for the Current Picture. This field is used for 48-bit addressing.
6	31:15	Reserved
	Format:	MBZ
	14:13	Direct MV Read Buffer - Tiled Resource Mode
	Format:	U2
	For Media Surfaces: This field specifies the tiled resource mode.	
	Value	Name
	0h	TRMODE_NONE
	1h	TRMODE_TILEYF
	2h	TRMODE_TILEYS
	3h	Reserved
	12:11	Reserved
	Format:	MBZ



MFX_VC1_DIRECTMODE_STATE											
10	Direct MV Read Buffer - Memory Compression Mode Format: U1 Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter, Media Memory Compression section for more details.										
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode				
	Value	Name									
	0	Horizontal Compression Mode									
1	Vertical Compression Mode										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> Memory compression will be attempted for this surface.		Format:	Enable								
Format:	Enable										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">U2</td> </tr> </table> This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.		Format:	U2								
Format:	U2										
8:7	Direct MV Read Buffer Base Address for the Current Picture - Arbitration Priority Control Format: U2 This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.										
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
	Value	Name									
	00b	Highest priority									
	01b	Second highest priority									
10b	Third highest priority										
11b	Lowest priority										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">U6</td> </tr> </table> The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.		Format:	U6								
Format:	U6										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">U6</td> </tr> </table> The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.		Format:	U6								
Format:	U6										
0	Reserved										



MFX_VC1_PRED_PIPE_STATE

MFX_VC1_PRED_PIPE_STATE			
Source:	VideoCS		
Length Bias:	2		
<p>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 performing the actual pixel remapping calculations/process.</p>			
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:	OpCode
	26:24	Media Command Opcode	
		Default Value:	2h VC1_COMMON
		Format:	OpCode
	23:21	SubOpcode A	
Default Value:		0h	
Format:		OpCode	
20:16	SubOpcode B		
	Default Value:	1h	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	0004h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	
1	31:16	Reserved	
		Format:	MBZ



MFX_VC1_PRED_PIPE_STATE			
15:14	<p>vin_intensitycomp_Double_FWDen</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>for forward reference picture only, to enable top or/and bottom of the reference field enable for single compensation. For frame, may only need one bit. 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.</p>	Format:	U2
Format:	U2		
13:12	<p>vin_intensitycomp_Double_BWDen</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>for backward reference picture only, no double for backward 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.</p>	Format:	U2
Format:	U2		
11:10	<p>vin_intensitycomp_Single_FWDen</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>for forward reference picture only, to enable top or/and bottom of the reference field enable for single compensation. For frame, may only need one bit. 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.</p>	Format:	U2
Format:	U2		
9:8	<p>vin_intensitycomp_Single_BWDen</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>for backward reference picture only, no double for backward 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.</p>	Format:	U2
Format:	U2		
7:4	<p>Reference Frame Boundary Replication Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> </table> <p>This is a bit field with each bit indicating the corresponding picture's boundary replication mode. Bit 11: reference 3 Bit 10: reference 2 Bit 9: reference 1 Bit 8: reference 0 0 = progressive frame replication 1 = interlace frame replication This field is maintained and provided by driver for both long and short VC1 interface format.</p>	Format:	U4
Format:	U4		



MFV_VC1_PRED_PIPE_STATE				
	3:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
2	31:30	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	29:24	LumShift2- single - FWD <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6
	Format:	U6		
	23:22	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	21:16	LumShift1 - single - FWD <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6
	Format:	U6		
15:14	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
13:8	LumScale2 - single - FWD <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6	
Format:	U6			
7:6	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
5:0	LumScale1 - Single - FWD <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6	
Format:	U6			
3	31:30	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			



MFV_VC1_PRED_PIPE_STATE				
	29:24	<p>LumShift2- double - FWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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, wBitstreamPCElement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.</p>	Format:	U6
	Format:	U6		
	23:22	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	21:16	<p>LumShift1 - double - FWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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, wBitstreamPCElement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.</p>	Format:	U6
	Format:	U6		
	15:14	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
13:8	<p>LumScale2 - double - FWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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, wBitstreamPCElement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.</p>	Format:	U6	
Format:	U6			
7:6	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
5:0	<p>LumScale1 - double - FWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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, wBitstreamPCElement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.</p>	Format:	U6	
Format:	U6			
4	31:30	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			



MFV_VC1_PRED_PIPE_STATE				
5	29:24	<p>LumShift2- single - BWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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, wBitstreamPCElement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.</p>	Format:	U6
	Format:	U6		
	23:22	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	21:16	<p>LumShift1 - single - BWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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, wBitstreamPCElement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.</p>	Format:	U6
	Format:	U6		
	15:14	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
13:8	<p>LumScale2 - single - BWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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, wBitstreamPCElement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.</p>	Format:	U6	
Format:	U6			
7:6	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
5:0	<p>LumScale1 - Single - BWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>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, wBitstreamPCElement and wBitstreamFcodes parameters provided by the DXVA2 VC1 interface to the driver for each current picture.</p>	Format:	U6	
Format:	U6			
	31:30	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			



MFX_VC1_PRED_PIPE_STATE			
29:24	<p>LumShift2- double - BWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6
Format:	U6		
23:22	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
21:16	<p>LumShift1 - double -BWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6
Format:	U6		
15:14	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
13:8	<p>LumScale2 - double - BWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6
Format:	U6		
7:6	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
5:0	<p>LumScale1 - double - BWD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U6</td> </tr> </table> <p>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.</p>	Format:	U6
Format:	U6		



MFX_VP8_BSP_BUF_BASE_ADDR_STATE

MFX_VP8_BSP_BUF_BASE_ADDR_STATE			
Source:		VideoCS	
Length Bias:		2	
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Video Codec
		Format:	OpCode
	26:24	Media Command OpCode	
		Default Value:	4h VP8
		Format:	OpCode
	23:21	Sub Opcode A	
Default Value:		2h VP8 Common	
Format:		OpCode	
20:16	Sub Opcode B		
	Default Value:	3h MFX_VP8_BSP_BUF_BASE_ADDR_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Format:	=n	
	Value	Name	Description
	000h	Excludes DWord (0,1) [Default]	A special case to provide a dummy image state for stitch mode operation. In this case, fields in DW1 which is part of the dummy image state command are ignored by hardware."
	008h		Used for normal encode mode
1	31:6	Frame Header Base Addr	
		Format:	StreamInAddress[31:6] 64 bytes aligned buffer in linear format.
		48-bit Abs. Address StreamIn Surface	
		Note: The format is linear vs. tile for better performance.	



MFX_VP8_BSP_BUF_BASE_ADDR_STATE																
	5:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ															
2	31:16	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
15:0	Frame Header Base Addr - Upper Range															
3	31:9	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
	8:7	Frame Header Base Addr - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority				
	Value	Name														
	00b	Highest priority														
01b	Second highest priority															
10b	Third highest priority															
11b	Lowest priority															
6:5	for FrameHeaderBaseAddr - LLC/eLLC Cacheability Control (LeLLCCC) This is the field used in GT interface block to determine what type of access need to be generated to uncore. For the cases where the LeLLCCC is set, cacheable transaction are generated to enable LLC usage for particular stream. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Cacheable</td> <td>Use Cacheability Controls from page table / UC with Fence (if coherent cycle)</td> </tr> <tr> <td>01b</td> <td>UC</td> <td>Uncacheable - non-cacheable</td> </tr> <tr> <td>10b</td> <td>WT</td> <td>Writethrough</td> </tr> <tr> <td>11b</td> <td>WB</td> <td>Writeback</td> </tr> </tbody> </table>	Value	Name	Description	00b	Cacheable	Use Cacheability Controls from page table / UC with Fence (if coherent cycle)	01b	UC	Uncacheable - non-cacheable	10b	WT	Writethrough	11b	WB	Writeback
Value	Name	Description														
00b	Cacheable	Use Cacheability Controls from page table / UC with Fence (if coherent cycle)														
01b	UC	Uncacheable - non-cacheable														
10b	WT	Writethrough														
11b	WB	Writeback														
4:3	Frame Header Base Addr - Target Cache (TC) This field allows the choice of LLC vs eLLC for caching <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>eLLC Only - not snooped in GT</td> </tr> <tr> <td>01b</td> <td>LLC Only</td> </tr> <tr> <td>10b</td> <td>LLC/eLLC Allowed</td> </tr> <tr> <td>11b</td> <td>L3, LLC, eLLC Allowed</td> </tr> </tbody> </table>	Value	Name	00b	eLLC Only - not snooped in GT	01b	LLC Only	10b	LLC/eLLC Allowed	11b	L3, LLC, eLLC Allowed					
Value	Name															
00b	eLLC Only - not snooped in GT															
01b	LLC Only															
10b	LLC/eLLC Allowed															
11b	L3, LLC, eLLC Allowed															
	2	Reserved														



MFX_VP8_BSP_BUF_BASE_ADDR_STATE												
	1:0	<p>Frame Header Base Addr - Age for QUADLRU (AGE)</p> <p>This field allows the selection of AGE parameter for a given surface in LLC or eLLC. . If a particular allocation is done at youngest age ("3") it tends to stay longer in the cache as compared to older age allocations ("2", "1", or "0"). This option is given to driver to be able to decide which surfaces are more likely to generate HITs, hence need to be replaced least often in caches.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Good chance of generating hits</td> </tr> <tr> <td>01b</td> <td>Next good chance of generating hits</td> </tr> <tr> <td>10b</td> <td>Decent chance of generating hits</td> </tr> <tr> <td>11b</td> <td>Poor chance of generating hits</td> </tr> </tbody> </table>	Value	Name	00b	Good chance of generating hits	01b	Next good chance of generating hits	10b	Decent chance of generating hits	11b	Poor chance of generating hits
	Value	Name										
	00b	Good chance of generating hits										
	01b	Next good chance of generating hits										
	10b	Decent chance of generating hits										
11b	Poor chance of generating hits											
4	31:6	<p>Intermediate Buffer Base Addr</p> <table border="1"> <tr> <td>Format:</td> <td>StreamInAddress[31:6] 64 bytes aligned buffer in linear format</td> </tr> <tr> <td colspan="2">48-bit AbsAddr StreamIn Surface</td> </tr> <tr> <td colspan="2">Note:The format is linear vs. tile for better performance.</td> </tr> </table>	Format:	StreamInAddress[31:6] 64 bytes aligned buffer in linear format	48-bit AbsAddr StreamIn Surface		Note: The format is linear vs. tile for better performance.					
	Format:	StreamInAddress[31:6] 64 bytes aligned buffer in linear format										
48-bit AbsAddr StreamIn Surface												
Note: The format is linear vs. tile for better performance.												
5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
5	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
15:0	<p>Intermediate Buffer Base Addr - Upper Range</p>											
6	31:9	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
8:7	<p>Intermediate Buffer Base Addr - Arbitration Priority Control</p> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority	
Value	Name											
00b	Highest priority											
01b	Second highest priority											
10b	Third highest priority											
11b	Lowest priority											



MFX_VP8_BSP_BUF_BASE_ADDR_STATE																
6:5	<p>Intermediate Buffer Base Addr - LLC/eLLC Cacheability Control (LeLLCCC)</p> <p>This is the field used in GT interface block to determine what type of access need to be generated to uncore.</p> <p>For the cases where the LeLLCCC is set, cacheable transaction are generated to enable LLC usage for particular stream.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Cacheable</td> <td>Use Cacheability Controls from page table / UC with Fence (if coherent cycle)</td> </tr> <tr> <td>01b</td> <td>UC</td> <td>Uncacheable - non-cacheable</td> </tr> <tr> <td>10b</td> <td>WT</td> <td>Writethrough</td> </tr> <tr> <td>11b</td> <td>WB</td> <td>Writeback</td> </tr> </tbody> </table>	Value	Name	Description	00b	Cacheable	Use Cacheability Controls from page table / UC with Fence (if coherent cycle)	01b	UC	Uncacheable - non-cacheable	10b	WT	Writethrough	11b	WB	Writeback
	Value	Name	Description													
	00b	Cacheable	Use Cacheability Controls from page table / UC with Fence (if coherent cycle)													
	01b	UC	Uncacheable - non-cacheable													
	10b	WT	Writethrough													
11b	WB	Writeback														
4:3	<p>Intermediate Buffer Base Addr- Target Cache (TC)</p> <p>This field allows the choice of LLC vs. eLLC for caching</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 75%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>eLLC Only - not snooped in GT</td> </tr> <tr> <td>01b</td> <td>LLC Only</td> </tr> <tr> <td>10b</td> <td>LLC/eLLC Allowed</td> </tr> <tr> <td>11b</td> <td>L3, LLC, eLLC Allowed</td> </tr> </tbody> </table>	Value	Name	00b	eLLC Only - not snooped in GT	01b	LLC Only	10b	LLC/eLLC Allowed	11b	L3, LLC, eLLC Allowed					
	Value	Name														
	00b	eLLC Only - not snooped in GT														
	01b	LLC Only														
	10b	LLC/eLLC Allowed														
11b	L3, LLC, eLLC Allowed															
2	Reserved															
1:0	<p>Intermediate Buffer Base Addr- Age for QUADLRU (AGE)</p> <p>This field allows the selection of AGE parameter for a given surface in LLC or eLLC. .</p> <p>If a particular allocation is done at youngest age ("3") it tends to stay longer in the cache as compared to older age allocations ("2", "1", or "0").</p> <p>This option is given to driver to be able to decide which surfaces are more likely to generate HITs, hence need to be replaced least often in caches.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 75%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Good chance of generating hits</td> </tr> <tr> <td>01b</td> <td>Next good chance of generating hits</td> </tr> <tr> <td>10b</td> <td>Decent chance of generating hits</td> </tr> <tr> <td>11b</td> <td>Poor chance of generating hits</td> </tr> </tbody> </table>	Value	Name	00b	Good chance of generating hits	01b	Next good chance of generating hits	10b	Decent chance of generating hits	11b	Poor chance of generating hits					
	Value	Name														
	00b	Good chance of generating hits														
	01b	Next good chance of generating hits														
	10b	Decent chance of generating hits														
11b	Poor chance of generating hits															
7	31:0	<p>Intermediate Buffer Partition-1 Offset</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U32</td> </tr> </table>	Format:	U32												
		Format:	U32													
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>All Intermediate Buffer Partition-[i] Offset (i = 1 to 8) and Intermediate Buffer Max Size need to be cacheline aligned (64Byte aligned).</td> </tr> </tbody> </table>	Programming Notes	All Intermediate Buffer Partition-[i] Offset (i = 1 to 8) and Intermediate Buffer Max Size need to be cacheline aligned (64Byte aligned).														
Programming Notes																
All Intermediate Buffer Partition-[i] Offset (i = 1 to 8) and Intermediate Buffer Max Size need to be cacheline aligned (64Byte aligned).																
8	31:0	<p>Intermediate Buffer Partition-2 Offset</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U32</td> </tr> </table>	Format:	U32												
		Format:	U32													



MFX_VP8_BSP_BUF_BASE_ADDR_STATE		
9	31:0	Intermediate Buffer Partition-3 Offset
		Format: U32
10	31:0	Intermediate Buffer Partition-4 Offset
		Format: U32
11	31:0	Intermediate Buffer Partition-5 Offset
		Format: U32
12	31:0	Intermediate Buffer Partition-6 Offset
		Format: U32
13	31:0	Intermediate Buffer Partition-7 Offset
		Format: U32
14	31:0	Intermediate Buffer Partition-8 Offset
		Format: U32
15	31:0	Intermediate Buffer Max Size
		Format: U32
16	31:6	Final Frame Base Addr
		Format: StreamInAddress[31:6] 64 bytes aligned buffer in linear format
		48-bit AbsAddr StreamIn Surface
		Note: The format is linear vs. tile for better performance.
5:0	Reserved	Format: MBZ
17	31:16	Reserved
		Format: MBZ
18	31:9	Reserved
		Format: MBZ
18	8:7	Final Frame Base Addr - Arbitration Priority Control
		Value Name
		00b Highest priority
		01b Second highest priority
		10b Third highest priority
11b Lowest priority		



MFX_VP8_BSP_BUF_BASE_ADDR_STATE																
6:5	<p>Final Frame Base Addr - LLC/eLLC Cacheability Control (LeLLCCC)</p> <p>This is the field used in GT interface block to determine what type of access need to be generated to uncore.</p> <p>For the cases where the LeLLCCC is set, cacheable transaction are generated to enable LLC usage for particular stream.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Cacheable</td> <td>Use Cacheability Controls from page table / UC with Fence (if coherent cycle)</td> </tr> <tr> <td>01b</td> <td>UC</td> <td>Uncacheable - non-cacheable</td> </tr> <tr> <td>10b</td> <td>WT</td> <td>Writethrough</td> </tr> <tr> <td>11b</td> <td>WB</td> <td>Writeback</td> </tr> </tbody> </table>	Value	Name	Description	00b	Cacheable	Use Cacheability Controls from page table / UC with Fence (if coherent cycle)	01b	UC	Uncacheable - non-cacheable	10b	WT	Writethrough	11b	WB	Writeback
	Value	Name	Description													
	00b	Cacheable	Use Cacheability Controls from page table / UC with Fence (if coherent cycle)													
	01b	UC	Uncacheable - non-cacheable													
	10b	WT	Writethrough													
	11b	WB	Writeback													
	4:3	<p>Final Frame Base Addr - Target Cache (TC)</p> <p>This field allows the choice of LLC vs eLLC for caching</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 75%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>eLLC Only - not snooped in GT</td> </tr> <tr> <td>01b</td> <td>LLC Only</td> </tr> <tr> <td>10b</td> <td>LLC/eLLC Allowed</td> </tr> <tr> <td>11b</td> <td>L3, LLC, eLLC Allowed</td> </tr> </tbody> </table>	Value	Name	00b	eLLC Only - not snooped in GT	01b	LLC Only	10b	LLC/eLLC Allowed	11b	L3, LLC, eLLC Allowed				
		Value	Name													
		00b	eLLC Only - not snooped in GT													
		01b	LLC Only													
10b		LLC/eLLC Allowed														
11b	L3, LLC, eLLC Allowed															
2	Reserved															
1:0	<p>Final Frame Base Addr - Age for QUADLRU (AGE)</p> <p>This field allows the selection of AGE parameter for a given surface in LLC or eLLC. .</p> <p>If a particular allocation is done at youngest age ("3") it tends to stay longer in the cache as compared to older age allocations ("2", "1", or "0").</p> <p>This option is given to driver to be able to decide which surfaces are more likely to generate HITS, hence need to be replaced least often in caches.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 75%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Good chance of generating hits</td> </tr> <tr> <td>01b</td> <td>Next good chance of generating hits</td> </tr> <tr> <td>10b</td> <td>Decent chance of generating hits</td> </tr> <tr> <td>11b</td> <td>Poor chance of generating hits</td> </tr> </tbody> </table>	Value	Name	00b	Good chance of generating hits	01b	Next good chance of generating hits	10b	Decent chance of generating hits	11b	Poor chance of generating hits					
	Value	Name														
	00b	Good chance of generating hits														
	01b	Next good chance of generating hits														
	10b	Decent chance of generating hits														
11b	Poor chance of generating hits															
19	<p>31:6 Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ														
<p>5:0 FinalFrameByteOffset</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U6</td> </tr> </table> <p>Specify byte offset within a 64-byte cacheline where the bitstream should be inserted at.</p>	Format:	U6														
Format:	U6															



MFX_VP8_BSP_BUF_BASE_ADDR_STATE																
20	31:6	Streamout Base Addr Format: StreamInAddress[31:6] 64 bytes aligned buffer in linear format 48-bit AbsAddr StreamIn Surface Note: The format is linear vs. tile for better performance.														
		5:4 Streamout Base Addr - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority				
		Value	Name													
		00b	Highest priority													
	01b	Second highest priority														
	10b	Third highest priority														
	11b	Lowest priority														
	3	Reserved														
	2	Streamout Base Addr - Graphics Data Type (GFDT) Format: U1 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	Streamout Base Addr - Cacheability Control Format: U2 EnumeratedType This field controls cacheability in the mid-level cache (MLC) and last-level cache (LLC) <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>GTT Entry</td> <td>Use cacheability control bits from GTT entry</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Not LLC or MLC</td> <td>Data is not cached in LLC or MLC</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>LLC but not MLC</td> <td>Data is cached in LLC but not MLC</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Both LLC and MLC</td> <td>Data is cached in both LLC and MLC</td> </tr> </tbody> </table>	Value	Name	Description	00b	GTT Entry	Use cacheability control bits from GTT entry	01b	Not LLC or MLC	Data is not cached in LLC or MLC	10b	LLC but not MLC	Data is cached in LLC but not MLC	11b	Both LLC and MLC
Value	Name	Description														
00b	GTT Entry	Use cacheability control bits from GTT entry														
01b	Not LLC or MLC	Data is not cached in LLC or MLC														
10b	LLC but not MLC	Data is cached in LLC but not MLC														
11b	Both LLC and MLC	Data is cached in both LLC and MLC														
21	31:16	Reserved Format: MBZ														
	15:0	Streamout Base Addr - Upper Range														
22	31:9	Reserved Format: MBZ														



MFX_VP8_BSP_BUF_BASE_ADDR_STATE		
8:7	Streamout Base Addr - Arbitration Priority Control	
	Value	Name
	00b	Highest priority
	01b	Second highest priority
	11b	Lowest priority
6:5	Streamout Base Addr - LLC/eLLC Cacheability Control (LeLLCCC)	
	This is the field used in GT interface block to determine what type of access need to be generated to uncore.	
	For the cases where the LeLLCCC is set, cacheable transaction are generated to enable LLC usage for particular stream.	
	Value	Name
	00b	Cacheable
	01b	UC
10b	WT	
11b	WB	
4:3	Streamout Base Addr - Target Cache (TC)	
	This field allows the choice of LLC vs eLLC for caching	
	Value	Name
	00b	eLLC Only - not snooped in GT
	01b	LLC Only
2	Reserved	
	1:0	Streamout Base Addr - Age for QUADLRU (AGE)
This field allows the selection of AGE parameter for a given surface in LLC or eLLC. .		
If a particular allocation is done at youngest age ("3") it tends to stay longer in the cache as compared to older age allocations ("2", "1", or "0").		
This option is given to driver to be able to decide which surfaces are more likely to generate HITS, hence need to be replaced least often in caches.		
Value		Name
00b	Good chance of generating hits	
01b	Next good chance of generating hits	
10b	Decent chance of generating hits	
11b	Poor chance of generating hits	



MFX_VP8_BSP_BUF_BASE_ADDR_STATE			
23	31:6	Coeff Probs StreamIn Surface	
		Format: StreamInAddress[31:6] 64 bytes aligned buffer in linear format	
		48-bit AbsAddr StreamIn Surface	
		Note: The format is linear vs. tile for better performance.	
	5:4	Coeff Probs StreamIn Surface - 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
	3	Reserved	
2		Coeff Probs StreamIn Surface - Graphics Data Type (GFDT)	
		Format: U1	
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	Coeff Probs StreamIn Surface - Cacheability Control		
	Format: U2 EnumeratedType		
	This field controls cacheability in the mid-level cache (MLC) and last-level cache (LLC)		
	Value	Name	
	Description		
	00b	GTT Entry	Use cacheability control bits from GTT entry
01b	Not LLC or MLC	Data is not cached in LLC or MLC	
10b	LLC but not MLC	Data is cached in LLC but not MLC	
11b	Both LLC and MLC	Data is cached in both LLC and MLC	
24	31:16	Reserved	
		Format: MBZ	
15:0	Coeff Probs StreamIn Surface - Upper Range		
	This field is for the upper range of Coeff Probs[4][8][3][11]		
25	31:9	Reserved	
		Format: MBZ	



MFX_VP8_BSP_BUF_BASE_ADDR_STATE			
8:7	Coeff Probs StreamIn Surface - Arbitration Priority Control		
	Value	Name	
	00b	Highest priority	
	01b	Second highest priority	
	10b	Third highest priority	
11b	Lowest priority		
6:5	Coeff Probs StreamIn Surface - LLC/eLLC Cacheability Control (LeLLCCC)		
	This is the field used in GT interface block to determine what type of access need to be generated to uncore.		
	For the cases where the LeLLCCC is set, cacheable transaction are generated to enable LLC usage for particular stream.		
	Value	Name	Description
	00b	Cacheable	Use Cacheability Controls from page table / UC with Fence (if coherent cycle)
	01b	UC	Uncacheable - non-cacheable
10b	WT	Writethrough	
11b	WB	Writeback	
4:3	Coeff Probs StreamIn Surface - Target Cache (TC)		
	This field allows the choice of LLC vs eLLC for caching		
	Value	Name	
	00b	eLLC Only - not snooped in GT	
	01b	LLC Only	
10b	LLC/eLLC Allowed		
11b	L3, LLC, eLLC Allowed		
2	Reserved		
1:0	Coeff Probs StreamIn Surface - Age for QUADLRU (AGE)		
	This field allows the selection of AGE parameter for a given surface in LLC or eLLC. .		
	If a particular allocation is done at youngest age ("3") it tends to stay longer in the cache as compared to older age allocations ("2", "1", or "0").		
	This option is given to driver to be able to decide which surfaces are more likely to generate HITS, hence need to be replaced least often in caches.		
	Value	Name	
	00b	Good chance of generating hits	
01b	Next good chance of generating hits		
10b	Decent chance of generating hits		
11b	Poor chance of generating hits		



MFX_VP8_BSP_BUF_BASE_ADDR_STATE																
26	31:6	Token Statistics Surface Format: TokenStatisticsAddress[31:6]														
		48-bit Abs. Address StreamIn Surface														
		Note: The format is linear vs. tile for better performance.														
		Frame Header Base Addr - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority				
	Value	Name														
00b	Highest priority															
01b	Second highest priority															
10b	Third highest priority															
11b	Lowest priority															
3	Reserved															
27	2	Token Statistics Surface - Graphics Data Type (GFDT) Format: U1 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	Token Statistics Surface - Cacheability Control This field controls cacheability in the mid-level cache (MLC) and last-level cache (LLC). <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>GTT Entry</td> <td>Use cacheability control bits from GTT entry</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Not LLC or MLC</td> <td>Data is not cached in LLC or MLC</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>LLC but not MLC</td> <td>Data is cached in LLC but not MLC</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Both LLC and MLC</td> <td>Data is cached in both LLC and MLC</td> </tr> </tbody> </table>	Value	Name	Description	00b	GTT Entry	Use cacheability control bits from GTT entry	01b	Not LLC or MLC	Data is not cached in LLC or MLC	10b	LLC but not MLC	Data is cached in LLC but not MLC	11b	Both LLC and MLC
Value	Name	Description														
00b	GTT Entry	Use cacheability control bits from GTT entry														
01b	Not LLC or MLC	Data is not cached in LLC or MLC														
10b	LLC but not MLC	Data is cached in LLC but not MLC														
11b	Both LLC and MLC	Data is cached in both LLC and MLC														
27	31:16	Reserved Format: MBZ														
	15:0	Token Statistics Surface This field is for the upper range of Token Statistics Address.														
28	31:9	Reserved Format: MBZ														



MFX_VP8_BSP_BUF_BASE_ADDR_STATE

8:7	Token Statistics Surface - 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 priority	
	11b	Lowest priority	
6:5	Memory Type: LLC/eLLC Cacheability Control (LeLLCCC) for CoeffProbs StreamIn Surface This is the field used in GT interface block to determine what type of access need to be generated to uncore. For the cases where the LeLLCCC is set, cacheable transaction are generated to enable LLC usage for particular stream.		
	Value	Name	Description
	00b	Use Cacheability Controls from page table / UC with Fence (if coherent cycle)	
	01b	UC	Uncacheable - non-cacheable
	10b	WT	Writethrough
	11b	WB	Writeback
4:3	Token Statistics Surface - Target Cache (TC) This field allows the choice of LLC vs eLLC for caching.		
	Value	Name	
	00b	eLLC Only - not snooped in GT	
	01b	LLC Only	
	10b	LLC/eLLC Allowed	
	11b	L3, LLC, eLLC Allowed	
2	Reserved		
1:0	Token Statistics Surface - Age for QUADLRU (AGE) This field allows the selection of AGE parameter for a given surface in LLC or eLLC. . If a particular allocation is done at youngest age ("3") it tends to stay longer in the cache as compared to older age allocations ("2", "1", or "0"). This option is given to driver to be able to decide which surfaces are more likely to generate HITs, hence need to be replaced least often in caches.		
	Value	Name	
	11b	Good chance of generating hits.	
	10b	Next good chance of generating hits	
	01b	Decent chance of generating hits	
	00b	Poor chance of generating hits	



MFX_VP8_BSP_BUF_BASE_ADDR_STATE																
29..30	63:6	<p>MPC RowStore Surface Address Low</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-6</td> </tr> </table> <p>Abs. Address StreamIn/StreamOut Surface. Note: The format is linear vs. tile for better performance.</p> <p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and MBZ.</p>	Format:	GraphicsAddress63-6												
	Format:	GraphicsAddress63-6														
	5:4	<p>MPC RowStore Base Addr - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority				
	Value	Name														
	00b	Highest priority														
01b	Second highest priority															
10b	Third highest priority															
11b	Lowest priority															
3	<p>Reserved</p>															
2	<p>MPC RowStore Surface Graphics Data Type (GFDT)</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U1</td> </tr> </table> <p>This field contains the GFDT bit for this surface when writes occur. GFDT can also be set by the GTT.</p> <p>The effective GFDT is the logical OR of this field with the GFDT from the GTT entry. This field is ignored for reads.</p>	Format:	U1													
Format:	U1															
1:0	<p>MPC RowStore Surface - Cacheability Control This field controls cacheability in the mid-level cache (MLC) and last-level cache (LLC).</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 25%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>GTT Entry</td> <td>Use cacheability control bits from GTT entry</td> </tr> <tr> <td>01b</td> <td>Not LLC or MLC</td> <td>Data is not cached in LLC or MLC</td> </tr> <tr> <td>10b</td> <td>LLC but not MLC</td> <td>Data is cached in LLC but not MLC</td> </tr> <tr> <td>11b</td> <td>Both LLC and MLC</td> <td>Data is cached in both LLC and MLC</td> </tr> </tbody> </table>	Value	Name	Description	00b	GTT Entry	Use cacheability control bits from GTT entry	01b	Not LLC or MLC	Data is not cached in LLC or MLC	10b	LLC but not MLC	Data is cached in LLC but not MLC	11b	Both LLC and MLC	Data is cached in both LLC and MLC
Value	Name	Description														
00b	GTT Entry	Use cacheability control bits from GTT entry														
01b	Not LLC or MLC	Data is not cached in LLC or MLC														
10b	LLC but not MLC	Data is cached in LLC but not MLC														
11b	Both LLC and MLC	Data is cached in both LLC and MLC														
31	31:9	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
8:7	<p>MPC RowStore - Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest Priority</td> </tr> <tr> <td>01b</td> <td>Second highest Priority</td> </tr> <tr> <td>10b</td> <td>Third highest Priority</td> </tr> <tr> <td>11b</td> <td>Lowest Priority</td> </tr> </tbody> </table>	Value	Name	00b	Highest Priority	01b	Second highest Priority	10b	Third highest Priority	11b	Lowest Priority					
Value	Name															
00b	Highest Priority															
01b	Second highest Priority															
10b	Third highest Priority															
11b	Lowest Priority															



MFX_VP8_BSP_BUF_BASE_ADDR_STATE

6:5	MPC RowStore - Memory Type: LLC/eLLC Cacheability Control	This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.	
	Value	Name	Description
	00b	Use Cacheability Controls from page table / UC with Fence (if coherent cycle)	
	01b	UC	Uncacheable - non-cacheable
	10b	WT	Writethrough
	11b	WB	Writeback
4:3	MPC RowStore - Target Cache		
	Format:	U2	
	This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.		
	Value	Name	
	00b	eLLC Only	
	01b	LLC Only	
	10b	LLC/eLLC Allowed	
	11b	L3, LLC, eLLC Allowed	
2	Reserved		
1:0	MPC RowStore Surface - Age for QUADLRU (AGE)	This field allows the selection of AGE parameter for a given surface in LLC or eLLC. . If a particular allocation is done at youngest age ("3") it tends to stay longer in the cache as compared to older age allocations ("2", "1", or "0"). This option is given to driver to be able to decide which surfaces are more likely to generate HITS, hence need to be replaced least often in caches.	
	Value	Name	
	11b	Good chance of generating hits.	
	10b	Next good chance of generating hits.	
	01b	Decent chance of generating hits.	
	00b	Poor chance of generating hits.	



MF8_VP8_Encoder_CFG

MF8_VP8_Encoder_CFG										
Source:	VideoCS									
Length Bias:	2									
This must be the very first command to issue after the surface state, the pipe select and base address setting commands and must be issued before MF8_VP8_PIC_STATE.										
DWord	Bit	Description								
0	31:29	Command Type								
		Default Value: 3h PARALLEL_VIDEO_PIPE								
		Format: OpCode								
	28:27	Pipeline								
		Default Value: 2h Video Codec								
		Format: OpCode								
	26:24	Media Command OpCode								
		Default Value: 4h VP8								
		Format: OpCode								
	23:21	Sub Opcode A								
Default Value: 2h VP8 Common										
Format: OpCode										
20:16	Sub Opcode B									
	Default Value: 1h MF8_VP8_ENCODER_CFG									
	Format: OpCode									
15:12	Reserved									
	Format: MBZ									
11:0	DWord Length	Format: =n								
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">000h</td> <td>Excludes DWord (0,1) [Default]</td> <td>A special case to provide a dummy image state for stitch mode operation. In this case, fields in DW1 which is part of the dummy image state command are ignored by hardware."</td> </tr> <tr> <td style="text-align: center;">01Dh</td> <td></td> <td>Used for normal encode mode</td> </tr> </tbody> </table>	Value	Name	Description	000h	Excludes DWord (0,1) [Default]	A special case to provide a dummy image state for stitch mode operation. In this case, fields in DW1 which is part of the dummy image state command are ignored by hardware."	01Dh	
	Value	Name	Description							
	000h	Excludes DWord (0,1) [Default]	A special case to provide a dummy image state for stitch mode operation. In this case, fields in DW1 which is part of the dummy image state command are ignored by hardware."							
	01Dh		Used for normal encode mode							
31:11	Reserved									
	Format: MBZ									
1	31:11	Reserved								
		Format: MBZ								



MFX_VP8_Encoder_CFG										
	10	VBSPunitPowerClock Gating Disable Format: <table border="1" style="display: inline-table; width: 150px;"><tr><td> </td><td>U1</td></tr></table> VBSPunit Power Clock Gating Disable.		U1						
		U1								
	9	Compressed Bitstream Output Disable Format: <table border="1" style="display: inline-table; width: 150px;"><tr><td> </td><td>U1</td></tr></table> Disable Compressed Bitstream Output. (Both Final Bitstream and Intermediate bit buffer)		U1						
		U1								
	8	Finer BRC Enable Format: <table border="1" style="display: inline-table; width: 150px;"><tr><td> </td><td>U1</td></tr></table> Enable Finer BRC Feature.		U1						
		U1								
	7	Per Segment Delta Qindex / LoopFilter Disable Format: <table border="1" style="display: inline-table; width: 150px;"><tr><td> </td><td>U1</td></tr></table> Disable Per Segment Delta Qindex / Loop Filter in Rate Control.		U1						
		U1								
	6	Rate Control Initial Pass Format: <table border="1" style="display: inline-table; width: 150px;"><tr><td> </td><td>U1</td></tr></table> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Initial pass</td> </tr> <tr> <td>0</td> <td>Subsequence Pass(es)</td> </tr> </tbody> </table>		U1	Value	Name	1	Initial pass	0	Subsequence Pass(es)
		U1								
Value	Name									
1	Initial pass									
0	Subsequence Pass(es)									
5	Skip Final Bitstream when Over / Under flow Format: <table border="1" style="display: inline-table; width: 150px;"><tr><td> </td><td>U1</td></tr></table> Skip Final Bitstream conditionally on Over/Under flow in rate control and intermediate Bit Buffer Overrun.		U1							
	U1									
4	Update Segment Feature Data Flag <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Exists If:</td> <td>//VP8 Encoder</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> Enable for Frame Header per Segment Quantizer / LoopFilter Update	Exists If:	//VP8 Encoder	Format:	U1					
Exists If:	//VP8 Encoder									
Format:	U1									
3	Bitstream Statistics Output Enable Enable Bitstream Statistics Output at Memory Surface in MFX_VBSP_BUF_ADDR_STATE DW[26:28]									
2	Token Statistics Output Enable Enable Token Statistics Output at Memory Surface in MFX_VBSP_BUF_ADDR_STATE DW[26:28]									



MFX_VP8_Encoder_CFG				
	1	Final Bitstream Output Disable <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> Disable Final Bitstream Output.	Format:	U1
	Format:	U1		
0	Performance Counter Enable <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> Enable Performance Counter in Streamout.	Format:	U1	
Format:	U1			
2	31:8	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
	7	Qindex_Clamp_High_mask for overflow <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> If current frame is overflow and this mask is set, it would mask out MFX_VP8_Img_Status register. DW1.bit1. In another word, subsequent passes would be skipped.	Format:	U1
Format:	U1			
	6	Qindex_Clamp_High_mask for underflow <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> If current frame is underflow and this mask is set, it would mask out MFX_VP8_Img_Status register. DW1.bit0. In another word, subsequent passes would be skipped	Format:	U1
Format:	U1			
	5	Final Bistream Buffer Overrun Enable Mask <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> Enable Final Bitstream Buffer Overrun detection feature.	Format:	U1
Format:	U1			
	4	Intermediate Bit Buffer Overrun Enable Mask <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> Enable Intermediate Bit Buffer Overrun detection feature.	Format:	U1
Format:	U1			
	3	Max Intra MB Bit Count Check Enable Mask <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> Enable Max. Intra MB bit count check in Streamout.	Format:	U1
Format:	U1			
	2	Max Inter MB Bit Count Check Enable Mask <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> Enable Max. Inter MB bit count check in Streamout.	Format:	U1
Format:	U1			



MFX_VP8_Encoder_CFG													
	1	<p>Min Frame Bit Count Rate Control Enable Mask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Enable Min. Frame Rate Control. This is a mask bit controlling if the condition of frame level bit count is less than or equal to FrameBitRateMin.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td></td> <td>If (Total Frame Level Bit Counter) = < (Frame Bit Rate Minimum limit) Set bit[0] and bit[1] of MFX_VP8_IMAGE_STATUS Control Register.</td> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>Do not update bit[0] of MFX_VP8_IMAGE_STATUS Control Register.</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	1		If (Total Frame Level Bit Counter) = < (Frame Bit Rate Minimum limit) Set bit[0] and bit[1] of MFX_VP8_IMAGE_STATUS Control Register.	0		Do not update bit[0] of MFX_VP8_IMAGE_STATUS Control Register.
	Format:	U1											
Value	Name	Description											
1		If (Total Frame Level Bit Counter) = < (Frame Bit Rate Minimum limit) Set bit[0] and bit[1] of MFX_VP8_IMAGE_STATUS Control Register.											
0		Do not update bit[0] of MFX_VP8_IMAGE_STATUS Control Register.											
	0	<p>Max Frame bit count Rate Control Enable Mask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Enable Max. Frame Rate Control. This is a mask bit controlling if the condition of frame level bit count is greater than or equal to FrameBitRateMax.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td></td> <td>If (Total Frame Level Bit Counter) >= (Frame Bit Rate Maximum Limit) Set bit[0] and bit[1] of MFX_VP8_IMAGE_STATUS control register.</td> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>Do not update bit[0] of MFX_VP8_IMAGE_STATUS control register.</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	1		If (Total Frame Level Bit Counter) >= (Frame Bit Rate Maximum Limit) Set bit[0] and bit[1] of MFX_VP8_IMAGE_STATUS control register.	0		Do not update bit[0] of MFX_VP8_IMAGE_STATUS control register.
Format:	U1												
Value	Name	Description											
1		If (Total Frame Level Bit Counter) >= (Frame Bit Rate Maximum Limit) Set bit[0] and bit[1] of MFX_VP8_IMAGE_STATUS control register.											
0		Do not update bit[0] of MFX_VP8_IMAGE_STATUS control register.											
3	31:28	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												
	27:16	<p>Max Intra MB Bit Count Limit</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U12</td> </tr> </table> <p>12-bit bit count for Max Intra MB Limit.</p>	Format:	U12									
Format:	U12												
	15:12	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												
	11:0	<p>Max Inter MB bit count</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U12</td> </tr> </table> <p>12-bit bit count for Max Inter MB Limit.</p>	Format:	U12									
Format:	U12												
4	31	<p>Frame Bitrate Min Unit Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>This field is the Frame Bitrate Minimum Limit Units.</p>	Format:	U1									
Format:	U1												



MFX_VP8_Encoder_CFG			
	Value	Name	Description
	0h	Compatibility Mode	Frame BitRate Min Unit is in old mode (128b/16Kb)
	1h	New Mode	Frame BitRate Min Unit is in new mode (32byte/4Kb)
30	Frame Bit Rate Min Unit		
	Format:		U1
	<i>This field is Frame Bitrate Minimum Mode.</i>		
	Value	Name	
	0	32-B	
	1	4-KB	
29:16	Frame Bit Rate Min		
	Format:		U14
	If either BRC Underflow or overflow is enabled. Frame Bit Rate Min and Frame Bit Rate Max need to be programmed with unambiguous values		
15	Frame Bitrate Max Unit Mode		
	Format:		U1
	This field is the Frame Bitrate Maximum Limit Units.		
	Value	Name	Description
	0h	Compatibility Mode	Frame BitRate Max Unit is in old mode (128b/16Kb)
	1h	New Mode	Frame BitRate Max Unit is in new mode (32byte/4Kb)
14	Frame Bit Rate Max Unit		
	Format:		U1
	<i>This field is Frame Bitrate Maximum Mode</i>		
	Value	Name	
	0	32-B	
	1	4-KB	
13:0	Frame Bit Rate Max		
	Format:		U14
	If either BRC Underflow or overflow is enabled. Frame Bit Rate Min and Frame Bit Rate Max need to be programmed with unambiguous values		



MFX_VP8_Encoder_CFG		
5	31:24	<p>Frame Delta QIndex Max[3] This field is the Frame level delta Qindex for total bit-count above FrameBitRateMax - First 1/8 Region. This field is used to calculate the suggested Frame Qindex into the MFX_VP8_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)].</p>
	23:16	<p>Frame DeltaQ Index Max[2] This field is the Frame level delta Qindex for bit-count above FrameBitRateMax - Above 1/8 and Below 1/4. This field is used to calculate the suggested Frame Qindex into the MFX_VP8_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)].</p>
	15:8	<p>Frame Delta QIndex Max[1] This field is the Frame level delta QINDEX for bit-count above FrameBitRateMax - Above 1/4 and Below 1/2. This field is used to calculate the suggested Frame QINDEX into the MFX_VP8_IMAGE_STATUS control register when total bit count for the entire frame is between 1/4 and 1/2 of FrameBitRateMaxDelta above FrameBitRateMax; i.e., In the range of [(FrameBitRateMax+ FrameBitRateMaxDelta » 2), (FrameBitRateMax+ FrameBitRateMaxDelta » 1)].</p>
	7:0	<p>Frame Delta QIndex Max [0] This field is the Frame level delta QINDEX for bit-count above FrameBitRateMax - Above 1/2. This field is used to calculate the suggested Frame QINDEX into the MFX_VP8_IMAGE_STATUS control register when total bit count for the entire frame is above FrameBitRateMax by more than half the distance of FrameBitRateMax; i.e., In the range of [(FrameBitRateMax + FrameBitRateMaxDelta » 1), ∞ (Infinite)].</p>
6	31:24	<p>Frame Delta QIndex Min[3] This field is the Frame level delta QINDEX for total bit-count below FrameBitRateMin - First 1/8 Region. This field is used to calculate the suggested Frame QINDEX into the MFX_VP8_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),</p>



MFX_VP8_Encoder_CFG		
		FrameBitRateMin].
	23:16	<p>Frame Delta QIndex Min[2] This field is the Frame level delta QINDEX for bit-count below FrameBitRateMin - Below 1/ 8 and Above 1/4. This field is used to calculate the suggested Frame QINDEX into the MFX_VP8_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 » 2), (FrameBitRateMin- FrameBitRateMinDelta » 3)].</p>
	15:8	<p>Frame Delta QIndex Min[1] This field is the Frame level delta QINDEX for bit-count below FrameBitRateMin- Below 1/4 and Above 1/2. This field is used to calculate the suggested Frame QINDEX into the MFX_VP8_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)].</p>
	7:0	<p>Frame Delta QIndex Min[0] This field is the Frame Level Delta QINDEX for bit-count below FrameBitRateMin - Below 1/2. This field is used to calculate the suggested Frame QINDEX into the MFX_VP8_IMAGE_STATUS control register when total bit count for the entire frame is below FrameBitRateMin by more than half the distance of FrameBitRateMin; i.e., In the range of [0, (FrameBitRateMin - FrameBitRateMinDelta » 1)].</p>
7	31:0	<p>Per Segment Frame Delta QIndex Max[1] Reserved</p>
8	31:0	<p>Per Segment Frame Delta QIndex Min[1] Reserved</p>
9	31:0	<p>Per Segment Frame Delta QIndex Max[2] Reserved</p>
10	31:0	<p>Per Segment Frame Delta QIndex Min[2] Reserved</p>
11	31:0	<p>Per Segment Frame Delta QIndex Max[3] Reserved</p>
12	31:0	<p>Per Segment Frame Delta QIndex Min[3] Reserved</p>



MFX_VP8_Encoder_CFG			
13	<p>31:24 Frame Delta Loop Filter Max[3]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>This field is the Frame level delta LoopFilter for total bit-count above FrameBitRateMax - First 1/8 region. This field is used to calculate the suggested Frame LoopFilter into the MFX_VP8_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)].</p>	Format:	U8
	Format:	U8	
	<p>23:16 Frame Delta Loop Filter Max[2]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>This field is the Frame level delta LoopFilter for bit-count above FrameBitRateMax - Above 1/8 and Below 1/4. This field is used to calculate the suggested Frame LoopFilter into the MFX_VP8_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) and (FrameBitRateMax + FrameBitRateMaxDelta » 2)].</p>	Format:	U8
	Format:	U8	
<p>15:8 Frame Delta Loop Filter Max[1]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>This field is the Frame level delta LOOPFILTER for bit-count above FrameBitRateMax - Above 1/4 and Below 1/2. This field is used to calculate the suggested Frame LOOPFILTER into the MFX_VP8_IMAGE_STATUS control register when total bit count for the entire frame is between 1/4 and 1/2 of FrameBitRateMaxDelta above FrameBitRateMax. i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta » 2) and (FrameBitRateMax+ FrameBitRateMaxDelta » 1)].</p>	Format:	U8	
Format:	U8		
<p>7:0 Frame Delta Loop Filter Max[0]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>This field is the Frame level delta LOOPFILTER for bit-count above FrameBitRateMax - Above 1/2. This field is used to calculate the suggested Frame LOOPFILTER into the MFX_VP8_IMAGE_STATUS control register when total bit count for the entire frame is above FrameBitRateMax by more than half the distance of FrameBitRateMaxDelta. i.e., in the range of ((FrameBitRateMax+ FrameBitRateMaxDelta » 1), infinite).</p>	Format:	U8	
Format:	U8		



MFX_VP8_Encoder_CFG				
14	31:24	Frame Delta Loop Filter Min[3] <table border="1" style="width: 100%;"><tr><td style="width: 70%;">Format:</td><td style="width: 30%;">U8</td></tr></table> <p>This field is the Frame level delta LOOPFILTER for total bit-count below FrameBitRateMin - First 1/8 region.</p> <p>This field is used to calculate the suggested Frame LOOPFILTER into the MFX_VP8_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 $[(\text{FrameBitRateMin} - \text{FrameBitRateMinDelta} \gg 3), \text{FrameBitRateMin})$.</p>	Format:	U8
	Format:	U8		
	23:16	Frame Delta Loop Filter Min[2] <table border="1" style="width: 100%;"><tr><td style="width: 70%;">Format:</td><td style="width: 30%;">U8</td></tr></table> <p>This field is the Frame level delta LOOPFILTER for bit-count below FrameBitRateMin - Below 1/ 8 and Above 1/4.</p> <p>This field is used to calculate the suggested Frame LOOPFILTER into the MFX_VP8_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 $[(\text{FrameBitRateMin} - \text{FrameBitRateMinDelta} \gg 2), (\text{FrameBitRateMin} - \text{FrameBitRateMinDelta} \gg 3))$.</p>	Format:	U8
	Format:	U8		
15:8	Frame Delta Loop Filter Min[1] <table border="1" style="width: 100%;"><tr><td style="width: 70%;">Format:</td><td style="width: 30%;">U8</td></tr></table> <p>This field is the Frame level delta LOOPFILTER for bit-count below FrameBitRateMin- Below 1/4 and Above 1/2.</p> <p>This field is used to calculate the suggested Frame LOOPFILTER into the MFX_VP8_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 $[(\text{FrameBitRateMin} - \text{FrameBitRateMinDelta} \gg 1) \text{ and } (\text{FrameBitRateMin} - \text{FrameBitRateMinDelta} \gg 2))$.</p>	Format:	U8	
Format:	U8			
7:0	Frame Delta Loop Filter Min[0] <table border="1" style="width: 100%;"><tr><td style="width: 70%;">Format:</td><td style="width: 30%;">U8</td></tr></table> <p>This field is the Frame Level Delta LOOPFILTER for bit-count below FrameBitRateMin - Below 1/ 2.</p> <p>This field is used to calculate the suggested Frame LOOPFILTER into the MFX_VP8_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 $[0, (\text{FrameBitRateMin} - \text{FrameBitRateMinDelta} \gg 1)$.</p>	Format:	U8	
Format:	U8			



MFX_VP8_Encoder_CFG			
15	31:0	Per Segment Frame Delta LoopFilter Max[1]	
16	31:0	Per Segment Frame Delta LoopFilter Min[1]	
17	31:0	Per Segment Frame Delta LoopFilter Max[2]	
18	31:0	Per Segment Frame Delta LoopFilter Min[2]	
19	31:0	Per Segment Frame Delta LoopFilter Max[3]	
20	31:0	Per Segment Frame Delta LoopFilter Min[3]	
21	31	Reserved	
	30:16	FrameBitRateMinDelta	
	Format:		U15
	This field is used to select the frame delta QINDEX when FrameBitRateMin Is exceeded. It shares the same FrameBitrateMinUnit.		
	Value	Name	Description
	[0-4095]		When FrameBitrateMinUnit is in Bytes, this range is in Bytes. When FrameBitrateMinUnit is in KB, this range is in KB units.
21	15	Reserved	
	14:0	Frame Bit Rate Max Delta	
	Format:		U15
	This field is used to select the frame delta QINDEX when FrameBitRateMax Is exceeded. It shares the same FrameBitrateMaxUnit.		
	Value	Name	Description
	[0-4095]		When FrameBitrateMinUnit is in Bytes, this range is in Bytes. When FrameBitrateMinUnit is in KB, this range is in KB units.
22	31:24	Reserved	
	Format:		MBZ
	23	Show Frame	
Format:		U1	
VP8 Frame Tag, Show Frame Field			



MFX_VP8_Encoder_CFG												
	22:20	Bitstream Format Version <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U3</td> </tr> </table> VP8 Frame Tag, Verison Field	Format:	U3								
	Format:	U3										
	19:18	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
17:16	Min Frame WSize Unit <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U2</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Compatibility Mode</td> <td>MinFrameWSizeUnit is in old mode (128b/16Kb)</td> </tr> <tr> <td>1h</td> <td>New Mode</td> <td>MinFrameWSizeUnit is in new mode (32byte/4Kb)</td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	Compatibility Mode	MinFrameWSizeUnit is in old mode (128b/16Kb)	1h	New Mode	MinFrameWSizeUnit is in new mode (32byte/4Kb)
Format:	U2											
Value	Name	Description										
0h	Compatibility Mode	MinFrameWSizeUnit is in old mode (128b/16Kb)										
1h	New Mode	MinFrameWSizeUnit is in new mode (32byte/4Kb)										
15:0	Min Frame WSize <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Exists If:</td> <td style="width: 40%;">//Encoder Only</td> </tr> </table> <p>This field (in Word, 16-bit) is specified to compensate for Intel® Rate Control.</p> <p>Zero padding would be performed.</p>	Exists If:	//Encoder Only									
Exists If:	//Encoder Only											
23	31:16	Vertical_Size_Code <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U16</td> </tr> </table> Frame Tag Vertical Size Code, composed of {VerticalScale[15:14], FrameHeight[13:0]}	Format:	U16								
	Format:	U16										
15:0	Horizontal_Size_Code <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U16</td> </tr> </table> Frame Tag Horizontal Size Code, composed of {HorizontalScale[15:14], FrameWidth[13:0]}	Format:	U16									
Format:	U16											
24	31:0	Frame Header Bit Count <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U32</td> </tr> </table> Binarized Header Bit Count.	Format:	U32								
Format:	U32											
25	31:0	Frame Header Bin Buffer Qindex Update Pointer <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U32</td> </tr> </table> Binarized Header Qindex Update Pointer If Segment Enabled and UpdateSegmentFeature enabled, 4 per segment Qindices would be updated in Binarized header (Only ABS mode supported). Else	Format:	U32								
Format:	U32											



MFX_VP8_Encoder_CFG											
		Base Qindex would be updated									
26	31:0	<p>Frame Header Bin Buffer LoopFilter Update Pointer</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Binarized Header LoopFilter Update Pointer If Segment Enabled and UpdateSegmentFeature enabled, 4 per segment LoopFilters would be updated in Binarized header (Only ABS mode supported). Else Base LoopFilter would be updated.</p>	Format:	U32							
Format:	U32										
27	31:0	<p>Frame Header Bin Buffer Token Update Pointer</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Binarized Header TokenUpdate Pointer</p>	Format:	U32							
Format:	U32										
28	31:0	<p>Frame Header Bin Buffer MVUpdate Pointer</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Binarized Header MVUpdate Pointer.</p>	Format:	U32							
Format:	U32										
29	31:28	ClampValues - CV7									
	27:24	CV6									
	23:20	CV5									
	19:16	CV4									
	15:12	CV3									
	11:8	CV2									
	7:4	CV1									
	3:0	<p>CV0 - Clamp Value 0</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>If the magnitude of coefficients at locations assigned with CV0 (mapping shown below) exceeds $2CV0-1$, they are replaced with $2CV0-1$. For coefficients at locations marked as 'none', no clamping is performed. The following mappings are only applied to luma and chroma blocks\subblocks containing AC coefficients (blocks\subblocks with only DC coeffs will not be clamped).</p> <p>For 4x4 frame block, each coefficient is mapped to one of the eight CV values as following:</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>none</td> <td>CV7</td> <td>CV5</td> <td>CV4</td> </tr> <tr> <td>CV7</td> <td>CV6</td> <td>CV4</td> <td>CV3</td> </tr> </table>	Format:	U4	none	CV7	CV5	CV4	CV7	CV6	CV4
Format:	U4										
none	CV7	CV5	CV4								
CV7	CV6	CV4	CV3								



MFX_VP8_Encoder_CFG

CV5	CV4	CV2	CV1
CV4	CV3	CV1	CV0

For 4x4 field block, each coefficient is mapped to one of the eight CV values as following:

none	CV6	CV3	CV1
CV7	CV6	CV3	CV1
CV5	CV4	CV2	CV0
CV5	CV4	CV2	CV0

Value	Name
0-15	



MFX_VP8_PAK_OBJECT

MFX_VP8_PAK_OBJECT			
Source:	VideoCS		
Length Bias:	2		
<p>The MFX_VP8_PAK_OBJECT command is the second primitive command for the VP8 Encoding Pipeline. The MV Data portion of the bitstream is loaded as indirect data object. Before issuing a MFX_VP8_PAK_OBJECT command, all VP8 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.</p> <p>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 first MB. MFX_VP8_PAK_OBJECT command follows the MbType definition like MFD.</p> <p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h MFX_VP8_PAK_OBJECT
		Format:	OpCode
	26:24	Media Command Opcode	
		Default Value:	4h VP8_ENC
Format:		OpCode	
23:21	SubOpcode A		
	Default Value:	2h	
	Format:	OpCode	
20:16	SubOpcode B		
	Default Value:	9h	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	5h DWORD_COUNT_n	
	Format:	=n Length -2	
1	31:30	Reserved	
		Format:	MBZ



MFX_VP8_PAK_OBJECT					
	29	<p>Enable Inline MV data</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field denotes if the MV data will be sent inline following the other inline data instead of being indirect.</p>	Format:	Enable	
	Format:	Enable			
	28:10	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ				
9:0	<p>Indirect PAK-MV Data Length</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U10</td> </tr> </table> <p>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 form). A value zero indicates that indirect data fetching is disabled - subsequently, the Indirect PAK-MV Data Start Address field is ignored. This field must have the same alignment as the Indirect PAK-MV Data Start Address. This field must be DW aligned (since each MV is 4 bytes in size). Driver has to derived this field from MVsize (MVquantity in DXVA, exact size) *4 bytes per MV.</p>	Format:	U10		
Format:	U10				
2	31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
	Format:	MBZ			
28:0	<p>Indirect PAK-MV Data Start Address Offset</p> <p>This field specifies the memory starting address (offset) of the MV data to be fetched into PAK Subsystem for processing. This pointer is relative to the MFC Indirect PAK-MV Object Base Address. Hardware ignores this field if indirect data is not present, i.e. the Indirect PAK-MV Data Length is set to 0. It is a Dword aligned address in all AVC encoding configuration, since each MV is 4 bytes in size.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 60%; text-align: center;">Value</th> <th style="width: 40%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,512MB)</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,512MB)	
Value	Name				
[0,512MB)					
3..6	127:0	<p>Inline Data</p> <p>All the required MB level controls and parameters for encoding are captured as Inline Data Description - VP8 PAK OBJECT. It has a fixed size of 4 DWs. Its definition is described in the next section.</p>			



MFX_VP8_PIC_STATE

MFX_VP8_PIC_STATE			
Source:	VideoCS		
Length Bias:	2		
This must be the very first command to issue after the surface state, the pipe select and base address setting commands and must be issued before MFX_VP8_IMG_STATE.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Video Codec
		Format:	OpCode
	26:24	Media Command OpCode	
		Default Value:	4h VP8
		Format:	OpCode
	23:21	Sub Opcode A	
Default Value:		0h VP8 Common	
Format:		OpCode	
20:16	Sub Opcode B		
	Default Value:	0h MFX_VP8_PIC_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length	Format:	=n
		Value	Name
	000h	Excludes DWord (0,1) [Default]	A special case to provide a dummy image state for stitch mode operation. In this case, fields in DW1 which is part of the dummy image state command are ignored by hardware."
	024h		Used for normal decode and encode mode
	1	31:24	Reserved
Exists If:			//Decoder / Encoder
Format:			MBZ



MFX_VP8_PIC_STATE															
	23:16 Frame Height Minus 1 <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Picture Height in integer number of MBs minus 1, so the min pic height can be program is 16 rows of pixels.</p>	Exists If:	//Decoder / Encoder	Format:	U8										
	Exists If:	//Decoder / Encoder													
	Format:	U8													
15:8 Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ											
Exists If:	//Decoder / Encoder														
Format:	MBZ														
	7:0 Frame Width Minus 1 <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Picture Width in integer number of MBs minus 1, so the min pic width can be program is 16 pixels.</p>	Exists If:	//Decoder / Encoder	Format:	U8										
	Exists If:	//Decoder / Encoder													
	Format:	U8													
2 31:26 Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ											
Exists If:	//Decoder / Encoder														
Format:	MBZ														
	25:24 Log2 Num of Partition <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>1 Token partition</td> </tr> <tr> <td style="text-align: center;">1</td> <td>2 Token partition</td> </tr> <tr> <td style="text-align: center;">2</td> <td>4 Token partition</td> </tr> <tr> <td style="text-align: center;">3</td> <td>8 Token partition</td> </tr> </tbody> </table>	Exists If:	//Decoder / Encoder	Format:	U2	Value	Name	0	1 Token partition	1	2 Token partition	2	4 Token partition	3	8 Token partition
	Exists If:	//Decoder / Encoder													
	Format:	U2													
	Value	Name													
	0	1 Token partition													
	1	2 Token partition													
2	4 Token partition														
3	8 Token partition														
23:19 Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ											
Exists If:	//Decoder / Encoder														
Format:	MBZ														
18:16 Deblock Sharpness Level <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Specify the sharpness level, as one of the regular deblocking strength control parameters.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Set to 0 to disable the use of sharpness control.</td> </tr> </tbody> </table>	Exists If:	//Decoder / Encoder	Format:	U3	Programming Notes	Set to 0 to disable the use of sharpness control.									
Exists If:	//Decoder / Encoder														
Format:	U3														
Programming Notes															
Set to 0 to disable the use of sharpness control.															



MFX_VP8_PIC_STATE		
15:14	Reserved	
	Exists If:	//Decoder / Encoder
	Format:	MBZ
13	Alternate Ref Pic MV SignBias Flag	
	Exists If:	//Decoder / Encoder
	Alternate Reference Picture MV sign bias flag, specified for non-key frame only.	
12	Golden Ref Picture MV SignBias Flag	
	Exists If:	//Decoder / Encoder
	Golden Reference Picture MV sign bias flag, specified for non-key frame only.	
11	Mode Reference Loop Filter Delta Enabled	
	Exists If:	//Decoder / Encoder
	Value	Name Description
	0	Mode or Reference Loop Filter Delta Adjustment for current frame is disabled.
	1	Mode or Reference Loop Filter Delta Adjustment for current frame is enabled.
10	MB NoCoeff SkipFlag	
	Exists If:	//Decoder / Encoder
	Frame level control if Skip MB (with no non-zero coefficient) is allowed or not.	
	Value	Name Description
	0	All MBs will have its MB level signaling mb_skip_coeff forced to 0. That is, no skip of coefficient record in the bitstream (even their values are all 0s)
	1	Skip MB is enabled in the per MB record.
9	Update MBSegment Map Flag	
	Exists If:	//Decoder / Encoder
	Value	Name Description
	0	Disable segmentation update
	1	Enable segmentation update, and to enable reading segment_id for each MB.
8	Segment Enable Flag	
	Exists If:	//Decoder / Encoder
	Value	Name Description
	0	Disable Segmentation processing in the current frame
	1	Enable Segmentation processing in the current frame



MFX_VP8_PIC_STATE			
7	Segmentation ID StreamIn Enable		
	Exists If:	//Decoder Only	
	Value	Name	
	0	StreamIn Disabled	
	1	StreamIn Enabled	
	Programming Notes		
	When 0, no input needed.		
	7:6	Reserved	
		Exists If:	//Encoder Only
		Format:	MBZ
6	Segmentation ID StreamOut Enable		
	Exists If:	//Decoder Only	
	Value	Name	
	0	StreamOut Disabled	
	1	StreamOut Enabled	
	Programming Notes		
	When 0, no output needed.		
	5	sKeyFrameFlag	
		Exists If:	//Decoder / Encoder
		Value	Name
0		Non-Key Frame (P-Frame)	
1		Key Frame (I-Frame)	
4	DBLKFilterType		
	Exists If:	//Decoder / Encoder	
	To specify VP8 Profile of operation.		
	Value	Name Description	
	0		Use a full feature normal deblocking filter
	1		Use a simple filter for deblocking
3:2	Reserved		
	Exists If:	//Decoder / Encoder	
	Format:	MBZ	



MFX_VP8_PIC_STATE			
1	Chroma Full Pixel MC Filter Mode		
	Exists If:	//Decoder / Encoder	
To specify VP8 Profile of operation.			
	Value	Name Description	
	0	Chroma MC filter operates in sub-pixel mode	
	1	Chroma MC filter only operates in full pixel position, i.e. no sub-pixel interpolation.	
0	MC Filter Select		
	Exists If:	//Decoder / Encoder	
To specify VP8 Profile of operation.			
	Value	Name Description	
	0	6-tap filter (regular filter mode)	
	1	2-tap bilinear filter (simple profile/version mode)	
3	31:30 Reserved		
	Exists If:	//Decoder / Encoder	
	Format:	MBZ	
	29:24 DBLKFilterLevel for Segment3		
	Exists If:	//Decoder / Encoder	
	Format:	U6	
		Value	Name Description
		0	Signifies disable in loop deblocking operation This is used to set a VP8 profile without in loop deblocker.
	Programming Notes		
	There are max 4 segments per frame, each segment can have its own deblocking filter level. When segmentation is disabled, only segment 0 parameter is used for the entire frame.		
	23:22 Reserved		
	Exists If:	//Decoder / Encoder	
Format:	MBZ		
21:16	DBLKFilterLevel for Segment2		
	Exists If:	//Decoder / Encoder	
	Format:	U6	
		Value	Name Description
	0	Signifies disable in loop deblocking operation This is used to set a VP8 profile without in loop deblocker.	



MFX_VP8_PIC_STATE												
		<p style="text-align: center;">Programming Notes</p> <p>There are max 4 segments per frame, each segment can have its own deblocking filter level. When segmentation is disabled, only segment 0 parameter is used for the entire frame.</p>										
	15:14	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ						
Exists If:	//Decoder / Encoder											
Format:	MBZ											
	13:8	<p>DBLKFilterLevel for Segment1</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Signifies disable in loop deblocking operation</td> <td>This is used to set a VP8 profile without in loop deblocker.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>There are max 4 segments per frame, each segment can have its own deblocking filter level. When segmentation is disabled, only segment 0 parameter is used for the entire frame.</p>	Exists If:	//Decoder / Encoder	Format:	U6	Value	Name	Description	0	Signifies disable in loop deblocking operation	This is used to set a VP8 profile without in loop deblocker.
Exists If:	//Decoder / Encoder											
Format:	U6											
Value	Name	Description										
0	Signifies disable in loop deblocking operation	This is used to set a VP8 profile without in loop deblocker.										
	7:6	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ						
Exists If:	//Decoder / Encoder											
Format:	MBZ											
	5:0	<p>DBLKFilterLevel for Segment0</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Signifies disable in loop deblocking operation</td> <td>This is used to set a VP8 profile without in loop deblocker.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>There are max 4 segments per frame, each segment can have its own deblocking filter level. When segmentation is disabled, only segment 0 parameter is used for the entire frame.</p>	Exists If:	//Decoder / Encoder	Format:	U6	Value	Name	Description	0	Signifies disable in loop deblocking operation	This is used to set a VP8 profile without in loop deblocker.
Exists If:	//Decoder / Encoder											
Format:	U6											
Value	Name	Description										
0	Signifies disable in loop deblocking operation	This is used to set a VP8 profile without in loop deblocker.										
4	31	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Encoder Only	Format:	MBZ						
Exists If:	//Encoder Only											
Format:	MBZ											
	30:24	<p>Seg 3 Qindex</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>U7</td> </tr> </table> <p>Quantizer Value for Segment ID 3</p>	Exists If:	//Encoder Only	Format:	U7						
Exists If:	//Encoder Only											
Format:	U7											



MFX_VP8_PIC_STATE		
23	Reserved	
	Exists If: //Encoder Only	
	Format: MBZ	
	Seg 2 Qindex	
	Exists If: //Encoder Only	
	Format: U7	
	Quantizer Value for Segment ID 2	
	Reserved	
	Exists If: //Encoder Only	
	Format: MBZ	
15	Reserved	
	Exists If: //Encoder Only	
	Format: MBZ	
14:8	Seg 1 Qindex	
	Exists If: //Encoder Only	
	Format: U7	
Quantizer Value for Segment ID 1		
7	Reserved	
	Exists If: //Encoder Only	
	Format: MBZ	
31:0	Reserved	
	Exists If: //Decoder Only	
	Format: MBZ	
6:0	Seg 0 Qindex	
	Exists If: //Encoder Only	
	Format: U7	
	Quantizer Value for Segment ID 0.	
	Programming Notes	
This is the [Default] Qindex		
5	31:29	Reserved
		Exists If: //Encoder Only
		Format: MBZ
	28	UVac Qindex Delta Sign
		Exists If: //Encoder Only
Format: U1		
Sign of Quantization index delta for UVac		



MFX_VP8_PIC_STATE				
27:24	UVac QindexDelta			
	<table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Absolute Quantization index delta for UVac</p>	Exists If:	//Encoder Only	Format:
Exists If:	//Encoder Only			
Format:	U4			
23:21	Reserved			
	<table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Encoder Only	Format:
Exists If:	//Encoder Only			
Format:	MBZ			
20	UVdc Qindex Delta Sign			
	<table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Sign of Quantization index delta for UVdc</p>	Exists If:	//Encoder Only	Format:
Exists If:	//Encoder Only			
Format:	U1			
19:16	UVdc Qindex Delta			
	<table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Absolute Quantization index delta for UVdc</p>	Exists If:	//Encoder Only	Format:
Exists If:	//Encoder Only			
Format:	U4			
15:13	Reserved			
	<table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Encoder Only	Format:
Exists If:	//Encoder Only			
Format:	MBZ			
12	Y2ac Qindex Sign			
	<table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Sign of Quantization index delta for Y2ac</p>	Exists If:	//Encoder Only	Format:
Exists If:	//Encoder Only			
Format:	U1			
11:8	Y2ac Qindex Delta			
	<table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Absolute Quantization index delta for Y2ac</p>	Exists If:	//Encoder Only	Format:
Exists If:	//Encoder Only			
Format:	U4			
7:5	Reserved			
	<table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Encoder Only	Format:
Exists If:	//Encoder Only			
Format:	MBZ			



MFX_VP8_PIC_STATE		
	4	Y2ac Qindex Delta Sign
		Exists If: //Encoder Only
		Format: U1
		Sign of Quantization index delta for Y2dc This is the [Default] Qindex Delta Sign
	31:0	Reserved
		Exists If: //Decoder Only
		Format: MBZ
	3:0	Y2dc Qindex Delta
		Exists If: //Encoder Only
		Format: U4
		Absolute Quantization index delta for Y2dc This is the [Default] Qindex Delta
6	31:5	Reserved
		Exists If: //Encoder Only
		Format: MBZ
	4	Y1dc Qindex Delta Sign
		Exists If: //Encoder Only
		Format: U1
		Sign of Quantization index delta for Y1dc This is the [Default] Qindex Delta Sign
	31:0	Reserved
		Exists If: //Decoder Only
		Format: MBZ
3:0	Y1dc Qindex Delta	
	Exists If: //Encoder Only	
	Absolute Quantization index delta for Y1dc This is the [Default] Qindex Delta	
7	31:15	Reserved
		Exists If: //Encoder Only
		Format: MBZ



MFX_VP8_PIC_STATE						
	14:8	Clamp Qindex high <table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>U7</td> </tr> </table> <p>Maximum Clamp Value for Qindex used in quantization.</p>	Exists If:	//Encoder Only	Format:	U7
	Exists If:	//Encoder Only				
	Format:	U7				
	7	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Encoder Only	Format:	MBZ
Exists If:	//Encoder Only					
Format:	MBZ					
31:0	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder Only	Format:	MBZ	
Exists If:	//Decoder Only					
Format:	MBZ					
6:0	Clamp Qindex Low <table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>U7</td> </tr> </table> <p>Minimum Clamp Value for Qindex used in quantization.</p>	Exists If:	//Encoder Only	Format:	U7	
Exists If:	//Encoder Only					
Format:	U7					
8	31:25	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder Only	Format:	MBZ
	Exists If:	//Decoder Only				
	Format:	MBZ				
	24:16	Quantizer Value [1][BlockType3=UVAC] <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U9</td> </tr> </table> <p>Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]</p>	Exists If:	//Decoder Only	Format:	U9
	Exists If:	//Decoder Only				
Format:	U9					
15:9	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder Only	Format:	MBZ	
Exists If:	//Decoder Only					
Format:	MBZ					
31:0	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Encoder Only	Format:	MBZ	
Exists If:	//Encoder Only					
Format:	MBZ					
8:0	Quantizer Value [1][BlockType2=UVDC] <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> </table> <p>Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]</p>	Exists If:	//Decoder Only			
Exists If:	//Decoder Only					
9	31:25	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder Only	Format:	MBZ
	Exists If:	//Decoder Only				
Format:	MBZ					



MFX_VP8_PIC_STATE		
	24:16	Quantizer Value [1][BlockType5=Y2AC]
		Exists If: //Decoder Only
		Format: U9
		Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]
	15:9	Reserved
		Exists If: //Decoder Only
		Format: MBZ
		Reserved
	31:0	Reserved
		Exists If: //Encoder Only
		Format: MBZ
		Reserved
8:0	Quantizer Value [1][BlockType4=Y2DC]	
	Exists If: //Decoder Only	
	Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]	
	Reserved	
10	31:25	Reserved
		Exists If: //Decoder Only
		Format: MBZ
		Reserved
	24:16	Quantizer Value [2][BlockType1=Y1AC]
		Exists If: //Decoder Only
		Format: U9
		Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]
	15:9	Reserved
		Exists If: //Decoder Only
		Format: MBZ
		Reserved
	31:0	Reserved
		Exists If: //Encoder Only
		Format: MBZ
		Reserved
	8:0	Quantizer Value [2][BlockType0=Y1DC]
		Exists If: //Decoder Only
		Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]
		Reserved
11	31:25	Reserved
		Exists If: //Decoder Only
		Format: MBZ
		Reserved



MFX_VP8_PIC_STATE		
	24:16	Quantizer Value [2][BlockType3=UVAC]
		Exists If: //Decoder Only
	Format: U9	
	Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]	
15:9	Reserved	
	Exists If: //Decoder Only	
Format: MBZ		
31:0	Reserved	
	Exists If: //Encoder Only	
Format: MBZ		
8:0	Quantizer Value [2][BlockType2=UVDC]	
	Exists If: //Decoder Only	
Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]		
12	31:25	Reserved
		Exists If: //Decoder Only
	Format: MBZ	
	24:16	Quantizer Value [2][BlockType5=Y2AC]
		Exists If: //Decoder Only
	Format: U9	
	Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]	
	15:9	Reserved
		Exists If: //Decoder Only
	Format: MBZ	
	31:0	Reserved
		Exists If: //Encoder Only
	Format: MBZ	
	8:0	Quantizer Value [2][BlockType4=Y2DC]
		Exists If: //Decoder Only
	Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]	
31:25	Reserved	
	Exists If: //Decoder Only	
Format: MBZ		



MFX_VP8_PIC_STATE			
	24:16	Quantizer Value [3][BlockType1=Y1AC]	
		Exists If: //Decoder Only	
		Format: U9	
			Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]
	15:9	Reserved	
		Exists If: //Decoder Only	
		Format: MBZ	
	31:0	Reserved	
		Exists If: //Encoder Only	
		Format: MBZ	
	8:0	Quantizer Value [3][BlockType0=Y1DC]	
		Exists If: //Decoder Only	
Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]			
14	31:25	Reserved	
		Exists If: //Decoder Only	
		Format: MBZ	
	24:16	Quantizer Value [3][BlockType3=UVAC]	
		Exists If: //Decoder Only	
		Format: U9	
			Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]
	15:9	Reserved	
		Exists If: //Decoder Only	
		Format: MBZ	
	31:0	Reserved	
		Exists If: //Encoder Only	
		Format: MBZ	
	8:0	Quantizer Value [3][BlockType2=UVDC]	
		Exists If: //Decoder Only	
		Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]	
	15	31:25	Reserved
			Exists If: //Decoder Only
			Format: MBZ



MFX_VP8_PIC_STATE		
	24:16	Quantizer Value [3][BlockType5=Y2AC]
		Exists If: //Decoder Only
		Format: U9
		Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]
	15:9	Reserved
		Exists If: //Decoder Only
		Format: MBZ
	31:0	Reserved
		Exists If: //Encoder Only
		Format: MBZ
	8:0	Quantizer Value [3][BlockType4=Y2DC]
		Exists If: //Decoder Only
		Quantizer Value [n = Segment_Id = 0..3][BlockType = 0..5]
16	31:6	CoeffProbability StreamIn Base Address
		Exists If: //Decoder Only
		Format: StreamInAddress[31:6] 64 bytes aligned buffer in linear format. (not tile for better performance)
		It is specified for non-key frame only. It is the final computed probability table for parsing Coeff in the bitstream. The buffer is unsigned 8-bit * 1056 entries (CoeffProbs[4][8][3][11].
	31:0	Reserved
		Exists If: //Encoder Only
		Format: MBZ
	5:0	Reserved
		Exists If: //Decoder Only
		Format: MBZ
17	31:16	Reserved
		Exists If: //Decoder Only
		Format: MBZ
	31:0	Reserved
		Exists If: //Encoder Only
		Format: MBZ



MFX_VP8_PIC_STATE																					
18	15:0	<p>CoeffProbability StreamIn Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Exists If:</td> <td>//Decoder Only</td> </tr> </table> <p>This field is for the upper range of CoeffProbability StreamIn Address</p>	Exists If:	//Decoder Only																	
	Exists If:	//Decoder Only																			
	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder Only	Format:	MBZ															
	Exists If:	//Decoder Only																			
	Format:	MBZ																			
	14:13	<p>CoeffProbability StreamIn - Tiled Resource Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Exists If:	//Decoder Only	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Exists If:	//Decoder Only																			
	Format:	U2																			
	Value	Name	Description																		
	0h	TRMODE_NONE	No tiled resource																		
	1h	TRMODE_TILEYF	4KB tiled resources																		
	2h	TRMODE_TILEYS	64KB tiled resources																		
	3h	Reserved																			
	12:11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder Only	Format:	MBZ															
	Exists If:	//Decoder Only																			
	Format:	MBZ																			
	10	<p>CoeffProbability StreamIn - Memory Compression Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter, Media Memory Compression section for more details.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 85%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td>1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Exists If:	//Decoder Only	Format:	U1	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode									
	Exists If:	//Decoder Only																			
Format:	U1																				
Value	Name																				
0	Horizontal Compression Mode																				
1	Vertical Compression Mode																				
9	<p>CoeffProbability StreamIn - Memory Compression Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Memory compression will be attempted for this surface.</p>	Exists If:	//Decoder Only	Format:	Enable																
Exists If:	//Decoder Only																				
Format:	Enable																				
31:7	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Encoder Only	Format:	MBZ																
Exists If:	//Encoder Only																				
Format:	MBZ																				



MFX_VP8_PIC_STATE																
19	8:7	<p>CoeffProbability StreamIn - Arbitration Priority Control</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Exists If:	//Decoder Only	Format:	U2	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
	Exists If:	//Decoder Only														
	Format:	U2														
	Value	Name														
	00b	Highest priority														
	01b	Second highest priority														
	10b	Third highest priority														
	11b	Lowest priority														
	6:1	<p>CoeffProbability StreamIn Address - Index to Memory Object Control State (MOCS) Tables</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Encoder Only</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>	Exists If:	//Encoder Only	Format:	U6										
	Exists If:	//Encoder Only														
Format:	U6															
0	Reserved															
31:24	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ											
Exists If:	//Decoder / Encoder															
Format:	MBZ															
23:16	<p>MBSegmentIDTreeProbs[2]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MBSegmentIDTreeProbs[2:0] probability tree table for CPBAC parsing Segment_ID of each MB.</p>	Exists If:	//Decoder / Encoder	Format:	U8											
Exists If:	//Decoder / Encoder															
Format:	U8															
15:8	<p>MBSegmentIDTreeProbs[1]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MBSegmentIDTreeProbs[2:0] probability tree table for CPBAC parsing Segment_ID of each MB.</p>	Exists If:	//Decoder / Encoder	Format:	U8											
Exists If:	//Decoder / Encoder															
Format:	U8															
7:0	<p>MBSegmentIDTreeProbs[0]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MBSegmentIDTreeProbs[2:0] probability tree table for CPBAC parsing Segment_ID of each MB.</p>	Exists If:	//Decoder / Encoder	Format:	U8											
Exists If:	//Decoder / Encoder															
Format:	U8															



		MFX_VP8_PIC_STATE	
20	31:24	MBNoCoeffSkipFalseProb	
		Exists If:	//Decoder / Encoder
		Format:	U8
	8-bit probability value for CPBAC parsing of the MBNoCoeffSkip Flag in the bistream.		
23:16		IntraMBProb	
		Exists If:	//Decoder / Encoder
		Format:	U8
	8-bit probability value for CPBAC parsing of the intra or inter MB type flag in the bitstream.		
15:8		InterPredFromLastRefProb	
		Exists If:	//Decoder / Encoder
		Format:	U8
	8-bit probability value for CPBAC parsing of the flag in the bitstream that determines which reference frame to be used for the current MB motion compensation.		
7:0		InterPredFromGRefRefProb	
		Exists If:	//Decoder / Encoder
		Format:	U8
	8-bit probability value for CPBAC parsing of the flag in the bitstream that determines which reference frame to be used for the current MB motion compensation.		
21	31:24	YModeProb[3]	
		Exists If:	//Decoder / Encoder
		Format:	U8
	YModeProb[3:0] probability tree table for CPBAC parsing Luma MBType of each MB.		
	23:16	YModeProb[2]	
		Exists If:	//Decoder / Encoder
		Format:	U8
	YModeProb[3:0] probability tree table for CPBAC parsing Luma MBType of each MB.		
	15:8	YModeProb[1]	
	Exists If:	//Decoder / Encoder	
	Format:	U8	
YModeProb[3:0] probability tree table for CPBAC parsing Luma MBType of each MB.			



MFX_VP8_PIC_STATE					
	7:0	YModeProb[0]			
		<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>YModeProb[3:0] probability tree table for CPBAC parsing Luma MBType of each MB.</p>	Exists If:	//Decoder / Encoder	Format:
Exists If:	//Decoder / Encoder				
Format:	U8				
22	31:24	Reserved			
		<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:
	Exists If:	//Decoder / Encoder			
	Format:	MBZ			
	23:16	UVModeProb[2]			
		<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>UVModeProb[2:0] probability tree table for CPBAC parsing Chroma MBType of each MB.</p>	Exists If:	//Decoder / Encoder	Format:
	Exists If:	//Decoder / Encoder			
	Format:	U8			
15:8	UVModeProb[1]				
	<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>UVModeProb[2:0] probability tree table for CPBAC parsing Chroma MBType of each MB.</p>	Exists If:	//Decoder / Encoder	Format:	U8
Exists If:	//Decoder / Encoder				
Format:	U8				
7:0	UVModeProb[0]				
	<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>UVModeProb[2:0] probability tree table for CPBAC parsing Chroma MBType of each MB.</p>	Exists If:	//Decoder / Encoder	Format:	U8
Exists If:	//Decoder / Encoder				
Format:	U8				
23	31:24	MVUpdateProbs[0][3]			
		<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:
	Exists If:	//Decoder / Encoder			
	Format:	U8			
23:16	MVUpdateProbs[0][2]				
	<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
Exists If:	//Decoder / Encoder				
Format:	U8				



MFX_VP8_PIC_STATE					
	<p>15:8 MVUpdateProbs[0][1]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder			
Format:	U8				
<p>7:0 MVUpdateProbs[0][0]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder				
Format:	U8				
24	<p>31:24 MVUpdateProbs[0][7]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder			
	Format:	U8			
	<p>23:16 MVUpdateProbs[0][6]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder			
Format:	U8				
<p>15:8 MVUpdateProbs[0][5]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder				
Format:	U8				
<p>7:0 MVUpdateProbs[0][4]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder				
Format:	U8				



MFX_VP8_PIC_STATE						
25	31:24	<p>MVUpdateProbs[0][11]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB.</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder				
	Format:	U8				
	23:16	<p>MVUpdateProbs[0][10]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB.</p>	Exists If:	//Decoder / Encoder	Format:	U8
Exists If:	//Decoder / Encoder					
Format:	U8					
15:8	<p>MVUpdateProbs[0][9]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB.</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					
7:0	<p>MVUpdateProbs[0][8]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB.</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					
26	31:24	<p>MVUpdateProbs[0][15]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder				
	Format:	U8				
23:16	<p>MVUpdateProbs[0][14]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					
15:8	<p>MVUpdateProbs[0][13]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					



MFX_VP8_PIC_STATE					
	7:0	MVUpdateProbs[0][12]			
		<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:
Exists If:	//Decoder / Encoder				
Format:	U8				
27	31:24	Reserved			
		<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:
	Exists If:	//Decoder / Encoder			
	Format:	MBZ			
	23:16	MVUpdateProbs[0][18]			
		<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:
	Exists If:	//Decoder / Encoder			
	Format:	U8			
	15:8	MVUpdateProbs[0][17]			
		<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:
	Exists If:	//Decoder / Encoder			
	Format:	U8			
7:0	MVUpdateProbs[0][16]				
	<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
Exists If:	//Decoder / Encoder				
Format:	U8				
28	31:24	MVUpdateProbs[1][3]			
		<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder Only	Format:
	Exists If:	//Decoder Only			
	Format:	U8			
23:16	MVUpdateProbs[1][2]				
	<table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder Only	Format:	U8
Exists If:	//Decoder Only				
Format:	U8				



MFX_VP8_PIC_STATE					
	<p>15:8 MVUpdateProbs[1][1]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder Only	Format:	U8
	Exists If:	//Decoder Only			
Format:	U8				
<p>7:0 MVUpdateProbs[1][0]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder Only	Format:	U8	
Exists If:	//Decoder Only				
Format:	U8				
29	<p>31:24 MVUpdateProbs[1][7]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder			
	Format:	U8			
	<p>23:16 MVUpdateProbs[1][6]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder			
Format:	U8				
<p>15:8 MVUpdateProbs[1][5]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder				
Format:	U8				
<p>7:0 MVUpdateProbs[1][4]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder				
Format:	U8				



MFX_VP8_PIC_STATE						
30	31:24	<p>MVUpdateProbs[1][11]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder				
	Format:	U8				
	23:16	<p>MVUpdateProbs[1][10]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
Exists If:	//Decoder / Encoder					
Format:	U8					
15:8	<p>MVUpdateProbs[1][9]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					
7:0	<p>MVUpdateProbs[1][8]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					
31	31:24	<p>MVUpdateProbs[1][15]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder				
Format:	U8					
23:16	<p>MVUpdateProbs[1][14]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					



MFX_VP8_PIC_STATE						
	15:8	MVUpdateProbs[1][13] <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
	Exists If:	//Decoder / Encoder				
Format:	U8					
7:0	MVUpdateProbs[1][12] <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					
32	31:24	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ
	Exists If:	//Decoder / Encoder				
	Format:	MBZ				
	23:16	MVUpdateProbs[1][18] <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8
Exists If:	//Decoder / Encoder					
Format:	U8					
15:8	MVUpdateProbs[1][17] <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					
7:0	MVUpdateProbs[1][16] <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>MVUpdateProbs[1:0][18:0] probability table for CPBAC parsing of MV update value of each MB. To map into DWord, it becomes MVUpdate[1:0][19:0].</p>	Exists If:	//Decoder / Encoder	Format:	U8	
Exists If:	//Decoder / Encoder					
Format:	U8					
33	31	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ
		Exists If:	//Decoder / Encoder			
Format:	MBZ					



MFX_VP8_PIC_STATE	
30:24	RefLFDelta3 (for ALTREF FRAME)
	Exists If: //Decoder / Encoder
	Format: S6 2's Compliment
	Delta value for reference frame based adjustment of the MB-level's filter level value. RefLFDeltas [ref_frametype = 0 to 3]
	Programming Notes
	Please note that although RefDelta is signed 2's complement, bitstream is sign bit + 6 bit magnitude.
23	Reserved
	Exists If: //Decoder / Encoder
	Format: MBZ
22:16	RefLFDelta2 (for GOLDEN FRAME)
	Exists If: //Decoder / Encoder
	Format: S6 2's Compliment
	Delta value for reference frame based adjustment of the MB-level's filter level value. RefLFDeltas [ref_frametype = 0 to 3]
	Programming Notes
	Please note that although RefDelta is signed 2's complement, bitstream is sign bit + 6 bit magnitude.
15	Reserved
	Exists If: //Decoder / Encoder
	Format: MBZ
14:8	RefLFDelta1 (for LAST FRAME)
	Exists If: //Decoder / Encoder
	Format: S6 2's Compliment
	Delta value for reference frame based adjustment of the MB-level's filter level value. RefLFDeltas [ref_frametype = 0 to 3]
	Programming Notes
	Please note that although RefDelta is signed 2's complement, bitstream is sign bit + 6 bit magnitude.



MFX_VP8_PIC_STATE	
7	Reserved
	Exists If: //Decoder / Encoder
	Format: MBZ
	6:0 RefLFDelta0 (for INTRA FRAME)
	Exists If: //Decoder / Encoder
	Format: S6 2's Compliment
	Delta value for reference frame based adjustment of the MB-level's filter level value. RefLFDeltas [ref_frametype = 0 to 3]
	Programming Notes
	Please note that although RefDelta is signed 2's complement, bitstream is sign bit + 6 bit magnitude.
	34
	Exists If: //Decoder / Encoder
	Format: MBZ
	30:24 ModelFDelta3 (for SPLITMV mode)
	Exists If: //Decoder / Encoder
	Format: S6 2's Compliment
	Delta value for mode based adjustment of the MB-level's filter level value. ModelFDeltas[MB_Type = 0 to 3]
	Programming Notes
	Please note that although ModelFDelta is signed 2's complement, bitstream is sign bit + 6 bit magnitude.
	23 Reserved
	Exists If: //Decoder / Encoder
	Format: MBZ
	22:16 ModelFDelta2 (for Nearest, Near and New mode)
	Exists If: //Decoder / Encoder
	Format: S6 2's Compliment
	Delta value for mode based adjustment of the MB-level's filter level value. ModelFDeltas[MB_Type = 0 to 3]
	Programming Notes



MFX_VP8_PIC_STATE					
	<p>Please note that although ModelFDelta is signed 2's complement, bitstream is sign bit + 6 bit magnitude.</p>				
15	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ
Exists If:	//Decoder / Encoder				
Format:	MBZ				
14:8	<p>ModelFDelta1 (for ZEROMV mode)</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>S6 2's Compliment</td> </tr> </table> <p>Delta value for mode based adjustment of the MB-level's filter level value. ModelFDeltas[MB_Type = 0 to 3]</p> <p style="text-align: center;">Programming Notes</p> <p>Please note that although ModelFDelta is signed 2's complement, bitstream is sign bit + 6 bit magnitude.</p>	Exists If:	//Decoder / Encoder	Format:	S6 2's Compliment
Exists If:	//Decoder / Encoder				
Format:	S6 2's Compliment				
7	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder / Encoder	Format:	MBZ
Exists If:	//Decoder / Encoder				
Format:	MBZ				
6:0	<p>ModelFDelta0 (for B_PRED mode)</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder / Encoder</td> </tr> <tr> <td>Format:</td> <td>S6 2's Compliment</td> </tr> </table> <p>Delta value for mode based adjustment of the MB-level's filter level value. ModelFDeltas[MB_Type = 0 to 3]</p> <p style="text-align: center;">Programming Notes</p> <p>Please note that although ModelFDelta is signed 2's complement, bitstream is sign bit + 6 bit magnitude.</p>	Exists If:	//Decoder / Encoder	Format:	S6 2's Compliment
Exists If:	//Decoder / Encoder				
Format:	S6 2's Compliment				
35	<p>31:0 Segmentation ID Stream Base Address</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>StreamAddress[31:0] 64 bytes linear aligned buffer</td> </tr> </table> <p>It is specified when SegmentationIDStreamInEnable or SegmentationIDStreamOutEnable is specified.</p> <p style="text-align: center;">Programming Notes</p> <p>Each cache has only 8 bits for 4 segmentation ID from 4 continuous MBs.</p>	Exists If:	//Decoder Only	Format:	StreamAddress[31:0] 64 bytes linear aligned buffer
Exists If:	//Decoder Only				
Format:	StreamAddress[31:0] 64 bytes linear aligned buffer				



MFX_VP8_PIC_STATE																				
36	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder Only	Format:	MBZ														
	Exists If:	//Decoder Only																		
Format:	MBZ																			
15:0	<p>Segmentation ID Stream Base Address [47:32]</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> </table> <p>This field is for the upper range of Segmentation ID Stream Base Address</p>	Exists If:	//Decoder Only																	
Exists If:	//Decoder Only																			
37	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder Only	Format:	MBZ														
	Exists If:	//Decoder Only																		
	Format:	MBZ																		
	14:13	<p>Segmentation ID Stream - Tiled Resource Mode</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Exists If:	//Decoder Only	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved
Exists If:	//Decoder Only																			
Format:	U2																			
Value	Name	Description																		
0h	TRMODE_NONE	No tiled resource																		
1h	TRMODE_TILEYF	4KB tiled resources																		
2h	TRMODE_TILEYS	64KB tiled resources																		
3h	Reserved																			
12:11	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Decoder Only	Format:	MBZ															
Exists If:	//Decoder Only																			
Format:	MBZ																			
10	<p>Segmentation ID Stream - Memory Compression Mode</p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Distinguishes Vertical from Horizontal compression. Please refer to vol1a Memory Data Formats chapter, Media Memory Compression section for more details.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Horizontal Compression Mode</td> </tr> <tr> <td>1</td> <td>Vertical Compression Mode</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	Horizontal Compression Mode	1	Vertical Compression Mode											
Format:	U1																			
Value	Name																			
0	Horizontal Compression Mode																			
1	Vertical Compression Mode																			
9	<p>Segmentation ID Stream - Memory Compression Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Memory compression will be attempted for this surface.</p>	Format:	Enable																	
Format:	Enable																			



MFX_VP8_PIC_STATE															
8:7	<p>Segmentation ID Stream - Arbitration Priority Control</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Exists If:</td> <td>//Decoder Only</td> </tr> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Highest priority</td> </tr> <tr> <td>01b</td> <td>Second highest priority</td> </tr> <tr> <td>10b</td> <td>Third highest priority</td> </tr> <tr> <td>11b</td> <td>Lowest priority</td> </tr> </tbody> </table>	Exists If:	//Decoder Only	Format:	U2	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
Exists If:	//Decoder Only														
Format:	U2														
Value	Name														
00b	Highest priority														
01b	Second highest priority														
10b	Third highest priority														
11b	Lowest priority														
6:1	<p>CoeffProbability StreamIn Address - Index to Memory Object Control State (MOCS) Tables</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U6</td> </tr> </table> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>	Format:	U6												
Format:	U6														
0	Reserved														



MFX_WAIT

MFX_WAIT		
Source:	VideoCS	
Length Bias:	1	
<p>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</p> <ul style="list-style-type: none"> • 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 pipeline. This 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
		Default Value: 01h MFX_SINGLE_DW
		Format: OpCode
	26:16	Sub-Opcode
		Default Value: 0h MFX_WAIT
		Format: OpCode
	15:10	Reserved
	Format: MBZ	
9	Reserved	
8	MFX Sync Control Flag	
	Description	
	If set, VCS will stall the parser until all prior MFX objects are completed down the MFX pipeline	
7:6	Reserved	
	Format: MBZ	
5:0	DWord Length	
	Default Value: 0h Excludes DWord (0,1)	
	Format: =n	
	Total Length - 2	



MI_ARB_CHECK

MI_ARB_CHECK			
Source:	VideoEnhancementCS		
Length Bias:	1		
Description			
<p>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.</p>			
Programming Notes			
<p>This instruction cannot be placed in a batch buffer.</p> <p>If execlist is enabled, there is a pending execution list and this command is parsed, then the command streamer will preempt the current context and start executing the new execution list.</p>			
DWord	Bit	Description	
0	31:29	MI Instruction Type	
		Default Value:	0h MI_INSTRUCTION
		Format:	OpCode
	28:23	MI Instruction Opcode	
		Default Value:	05h MI_ARB_CHECK
		Format:	OpCode
	22:0	Reserved	
		Format:	MBZ



MI_ARB_CHECK

MI_ARB_CHECK			
Source:	BlitterCS		
Length Bias:	1		
Description			
<p>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.</p>			
Programming Notes			
<p>This instruction cannot be placed in a batch buffer.</p> <p>If execlist is enabled, there is a pending execution list and this command is parsed, then the command streamer will preempt the current context and start executing the new execution list.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_INSTRUCTION
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	05h MI_ARB_CHECK
		Format:	OpCode
	22:0	Reserved	
		Format:	MBZ



MI_ARB_CHECK

MI_ARB_CHECK			
Source:	RenderCS		
Length Bias:	1		
Description			
<p>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.</p>			
Programming Notes			
<p>Ring Buffer mode of scheduling:</p> <ul style="list-style-type: none"> 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. <p>Execlist mode of scheduling: MI_ARB_CHK will be used to indicate a command boundary on which Preemption will be honored by Command Streamer in the execlist mode of operation. UHPTR is ignored when processing MI_ARB_CHK in execlist mode.</p> <p>This instruction can be in either a ring buffer or batch buffer.</p> <p>MI_ARB_CHK command must not be programmed in INDIRECT_CTX and BB_PER_CTX_PTR buffers.</p>			
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_ARB_CHECK

MI_ARB_CHECK			
Source:	VideoCS		
Length Bias:	1		
Description			
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 cannot be placed in a batch buffer.			
If execlist is enabled, there is a pending execution list and this command is parsed, then the command streamer will preempt the current context and start executing the new execution list.			
DWord	Bit	Description	
0	31:29	MI Instruction Type	
		Default Value:	0h MI_INSTRUCTION
		Format:	OpCode
	28:23	MI Instruction Opcode	
		Default Value:	05h MI_ARB_CHECK
		Format:	OpCode
22:0	Reserved		
	Format:	MBZ	



MI_ARB_ON_OFF

MI_ARB_ON_OFF		
Source:	CommandStreamer	
Length Bias:	1	
Description		
<p>The MI_ARB_ON_OFF instruction is used to disable/enable context switching. This instruction can be used to prevent submission of a new execlist from interrupting a command sequence, however lite restore preemption is allowed with in the arbitration disabled command execution zone. Note that context switching will remain disabled until re-enabled through use of this command.</p> <p>This command will also prevent a switch in the case of waiting on events, running out of commands. These will effectively hang the device if allowed to occur while arbitration is off (context switching is disabled.) This command should always be used as an off-on pair with the sequence of instructions to be protected from context switch between MI_ARB_OFF and MI_ARB_ON. Software must use this arbitration control with caution since it has the potential to increase the response time of the Render Engine to pre-emption requests. This is a privileged command; it will not be effective (will be converted to a no-op) if executed from within a non-privileged batch buffer.</p>		
<p>Execution List Mode of Scheduling: The MI_ARB_ON_OFF instruction is used to disable/enable context switching. Context swiching could be either due to preemption or un-succesfull wait for events or semaphore waits. This instruction can be used to prevent submission of a new execlist from interrupting a command sequence, however lite restore preemption is allowed with in the arbitration disabled command execution zone. Note that context switching will remain disabled until re-enabled through use of this command. This command will also prevent a switch in the case of waiting on events, running out of commands. These will effectively hang the device if allowed to occur while arbitration is off (context switching is disabled.)</p> <p>Ring Buffer Mode of Scheduling: The MI_ARB_ON_OFF instruction is used to disable preemption on the preemptable commands. SW can explicitly make section of commands in a command buffer non-preemptable by sandwiching them between ARB_OFF and ARB_ON, HW will ingore preemption request (UHPTR Valid) until arbitration is enabled.</p>		
Programming Notes		
<p>This command must be always be programmed in pairs of off/on in the same command dispatch. Sequence of instructions to be protected from cntext switch or preemption must be programmed between the MI_ARB_OFF and MI_ARB_ON. Software must use this arbitration control with caution since it has the potential to increase the response time of the Render Engine to pre-emption requests. This is a privileged command; it will not be effective (will be converted to a no-op) if executed from within a non-privileged batch buffer.</p>		
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		
Format: OpCode		



MI_ARB_ON_OFF			
	22:2	Reserved	
		Format: MBZ	
	1	Reserved	
		Source: BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS	
		Format: MBZ	
	0	Arbitration Enable	
		Format: Enable	
		This field enables or disables context switches due to pre-emption (a new execlist).	
		Value	Name
		1	Arbitration Enabled [Default]
0	Arbitration Disabled		



MI_ATOMIC

MI_ATOMIC											
Source:	BSpec										
Length Bias:	2										
Description											
<p>MI_ATOMIC is used to carry atomic operation on data in graphics memory. Atomic operations are supported on data granularity of 4B, 8B and 16B. The atomic operation leads to a read-modify-write operation on the data in graphics memory with the option of returning value. The data in graphics memory is modified by doing arithmetic and logical operation with the inline/indirect data provided with the MI_ATOMIC command. Inline/Indirect provided in the command can be one or two operands based on the atomic operation. Ex: Atomic-Compare operation needs two operands while Atomic-Add operation needs single operand and Atomic-increment requires no operand. Refer "Atomics" sub-section under "L3 Cache and URB" section for detailed atomic operations supported. Atomic operations can be enabled to return value by setting "Return Data Control" field in the command, return data is stored to CS_GPR registers. CS_GPR4/5 registers are updated with memory Return Data based on the "Data Size". Each GPR register is qword in size and occupies two MMIO registers.</p> <p>Note: Any references to CS_GPR registers in the command should be understood as the CS_GPR registers belonging to the corresponding engines *CS_GPR registers.</p> <table border="1"> <thead> <tr> <th>Engine Name</th> <th>Corresponding GPR Registers</th> </tr> </thead> <tbody> <tr> <td>RCS</td> <td>CS_GPR</td> </tr> <tr> <td>BCS</td> <td>BCS_GPR</td> </tr> <tr> <td>VCS</td> <td>VCS_GPR</td> </tr> <tr> <td>VECS</td> <td>VECS_GPR</td> </tr> </tbody> </table> <p>Indirect Source Operands: Operand1 is sourced from [CS_GPR1, CS_GPR0] Operand2 is sourced from [CS_GPR3, CS_GPR2] Read return Data is stored in [CS_GPR_5, CS_GPR4]</p> <p>When "Data Size" is DWORD lower dword of CS_GPR4 (Qword) is updated with the dword data returned from memory. When "Data Size" is QWORD only CS_GPR4 (Qword) is updated with the qword data returned from memory. When the data size is OCTWORD CS_GPR4/5 are updated with the OCTWORD data returned from memory. CS_GPR4 is loaded with lower qword returned from memory and CS_GPR5 is loaded with upper qword returned from memory.</p>		Engine Name	Corresponding GPR Registers	RCS	CS_GPR	BCS	BCS_GPR	VCS	VCS_GPR	VECS	VECS_GPR
Engine Name	Corresponding GPR Registers										
RCS	CS_GPR										
BCS	BCS_GPR										
VCS	VCS_GPR										
VECS	VECS_GPR										
Programming Notes											
<ul style="list-style-type: none"> When Inline Data mode is not set, Dwords 3..10 must not be included as part of the command. Dword Length field in the header must be programmed accordingly. When Inline Data Mode is set, Dwords 3..10 must be included based on the Data Size field of the header. Both Operand-1 and Operand-2 dwords must be programmed based on the Data Size field. Operand-2 must be programmed to 0x0 if the atomic operation doesn't require it. Dword Length field in the header must be programmed accordingly. 											



DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	0h MI_COMMAND	
		Format:	OpCode	
	28:23	MI Command Opcode		
		Default Value:	2Fh MI_ATOMIC	
		Format:	OpCode	
	22	Memory Type		
		This bit will be ignored and treated as if clear when executing from a non-privileged batch buffer. It is allowed for this bit to be clear when executing this command from a privileged (secure) batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear.		
		Value	Name	Description
		0h	Per Process Graphics Address	
1h		Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.	
21	Reserved			
	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS		
	Format:	MBZ		
21	Post-Sync Operation			
	Source:	RenderCS		
	Value	Name	Description	
	0h	No Post Sync Operation	Command is executed as usual.	
	1h	Post Sync Operation	MI_ATOMIC command is executed as a pipelined PIPE_CONTROL flush command with Atomics operation as post sync operation. Flush completion only guarantees the workload prior to this command is pushed till Windower unit and completion of any outstanding flushes issued prior to this command. When this bit set following restriction apply to atomic operation: <ul style="list-style-type: none"> Non-Compare atomic operations are supported on data granularity of 4B and 8B. DW3 is the lower dword of the operand and DW4 is the upper dword of the operand for the atomic operation. Compare atomic operations are supported on data granularity of 4B. DW3 is Operand-0 and DW4 is Operand-1 for the atomic operation. 	



MI_ATOMIC																	
		<ul style="list-style-type: none"> • Atomic operations to GGTT/PPGTT memory surface are supported. • Only Inline data mode for atomic operand is supported, no support for indirect data mode. • No support for Return Data Control functionality. • No support for atomic operations on data granularity of 16B. • No support for compare atomic operations on data granularity of 8B. 															
Programming Notes																	
Any desired pipeline flush operation can be achieved by programming PIPE_CONTROL command prior to this command.																	
AWhen this bit is set Command Streamer sends a flush down the pipe and the atomic operation is saved as post sync operation. Command streamer goes on executing the following commands. Atomic operation saved as post sync operation is executed at some point later on completion of corresponding flush issued.																	
AWhen this bit is set atomic semaphore signal operation will be out of order with rest of the MI commands programmed in the ring buffer or batch buffer, it will be in order with respect to the post sync operations resulting due to PIPE_CONTROL command.																	
20:19	Data Size	<p>This field indicates the size of the operand in dword/qword/octword on which atomic operation will be performed. Data size must match with the Atomic Opcode. Operation Data size could be 4B, 8B or 16B</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>DWORD</td> <td>Operand size used by Atomic Operation is DWORD.</td> </tr> <tr> <td>1h</td> <td>QWORD</td> <td>Operand Size used by Atomic Operation is QWORD.</td> </tr> <tr> <td>2h</td> <td>OCTWORD</td> <td>Operand Size used by Atomic Operation is OCTWORD.</td> </tr> <tr> <td>3h</td> <td>RESERVED</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	DWORD	Operand size used by Atomic Operation is DWORD.	1h	QWORD	Operand Size used by Atomic Operation is QWORD.	2h	OCTWORD	Operand Size used by Atomic Operation is OCTWORD.	3h	RESERVED	
Value	Name	Description															
0h	DWORD	Operand size used by Atomic Operation is DWORD.															
1h	QWORD	Operand Size used by Atomic Operation is QWORD.															
2h	OCTWORD	Operand Size used by Atomic Operation is OCTWORD.															
3h	RESERVED																
18	Inline Data	<p>This bit when set indicates the source operands are provided in line within the command. When reset the source operands are in CS_GPR registers.</p>															
Programming Notes																	
CS_GPR registers must be programmed with appropriate values before issuing MI_ATOMIC command with this field reset.																	



MI_ATOMIC													
17	<p>CS STALL</p> <p>This bit when set command stream waits for completion of this command before executing the next command.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> <th style="text-align: center;">Source</th> </tr> </thead> <tbody> <tr> <td>Render Command Streamer Only: CS will not guarantee atomic operation to be complete upon setting this bit along with Post Sync Operation set. When Post Sync Operation is set, this bit has no significance.</td> <td>RenderCS</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Workaround</th> </tr> </thead> <tbody> <tr> <td>Workaround : When CS STALL bit is set, Return Data Control must also be set in MI_ATOMIC command.</td> </tr> </tbody> </table>	Programming Notes	Source	Render Command Streamer Only: CS will not guarantee atomic operation to be complete upon setting this bit along with Post Sync Operation set. When Post Sync Operation is set, this bit has no significance.	RenderCS	Workaround	Workaround : When CS STALL bit is set, Return Data Control must also be set in MI_ATOMIC command.						
Programming Notes	Source												
Render Command Streamer Only: CS will not guarantee atomic operation to be complete upon setting this bit along with Post Sync Operation set. When Post Sync Operation is set, this bit has no significance.	RenderCS												
Workaround													
Workaround : When CS STALL bit is set, Return Data Control must also be set in MI_ATOMIC command.													
16	<p>Return Data Control</p> <table border="1" style="width: 100%;"> <tr> <td>Source:</td> <td>RenderCS, BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS</td> </tr> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>When Return Data Control is set the read return feature will be enabled during the atomic operation. Data is stored in CS_GPR5/4 registers unconditionally on completion of the atomic operation. On data return CS_GPR5/4 Registers are updated based on the "Data Size" field. When "Data Size" is DWORD lower dword of CS_GPR4 (Qword) is updated with the dword data returned from memory. When "Data Size" is QWORD only CS_GPR4 (Qword) is updated with the qword data returned from memory. When the data size is OCTWORD CS_GPR4/5 are updated with the OCTWORD data returned from memory. CS_GPR4 is loaded with lower qword returned from memory and CS_GPR5 is loaded with upper qword returned from memory.</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Workaround</th> </tr> </thead> <tbody> <tr> <td>Workaround: When Return Data Control bit is set, CS STALL must also be set in MI_ATOMIC command.</td> </tr> </tbody> </table>	Source:	RenderCS, BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS	Description	When Return Data Control is set the read return feature will be enabled during the atomic operation. Data is stored in CS_GPR5/4 registers unconditionally on completion of the atomic operation. On data return CS_GPR5/4 Registers are updated based on the "Data Size" field. When "Data Size" is DWORD lower dword of CS_GPR4 (Qword) is updated with the dword data returned from memory. When "Data Size" is QWORD only CS_GPR4 (Qword) is updated with the qword data returned from memory. When the data size is OCTWORD CS_GPR4/5 are updated with the OCTWORD data returned from memory. CS_GPR4 is loaded with lower qword returned from memory and CS_GPR5 is loaded with upper qword returned from memory.	Workaround	Workaround: When Return Data Control bit is set, CS STALL must also be set in MI_ATOMIC command.						
Source:	RenderCS, BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS												
Description													
When Return Data Control is set the read return feature will be enabled during the atomic operation. Data is stored in CS_GPR5/4 registers unconditionally on completion of the atomic operation. On data return CS_GPR5/4 Registers are updated based on the "Data Size" field. When "Data Size" is DWORD lower dword of CS_GPR4 (Qword) is updated with the dword data returned from memory. When "Data Size" is QWORD only CS_GPR4 (Qword) is updated with the qword data returned from memory. When the data size is OCTWORD CS_GPR4/5 are updated with the OCTWORD data returned from memory. CS_GPR4 is loaded with lower qword returned from memory and CS_GPR5 is loaded with upper qword returned from memory.													
Workaround													
Workaround: When Return Data Control bit is set, CS STALL must also be set in MI_ATOMIC command.													
15:8	<p>ATOMIC OPCODE</p> <p>This field selects the kind of atomic operation to be performed. Refer "Atomics" sub-section of "L3 Cache and URB" section of the B-spec for atomic opcode encoding and operation.</p>												
7:0	<p>DWord Length</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>=n</td> </tr> </table> <p>Total Length - 2. Excludes DWord (0,1).</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Exists If</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td>[Default]</td> <td>([Inline Data]=0)</td> </tr> <tr> <td>9h</td> <td></td> <td>([Inline Data]=1)</td> </tr> </tbody> </table>	Format:	=n	Value	Name	Exists If	1h	[Default]	([Inline Data]=0)	9h		([Inline Data]=1)	
Format:	=n												
Value	Name	Exists If											
1h	[Default]	([Inline Data]=0)											
9h		([Inline Data]=1)											



MI_ATOMIC				
1	31:2	<p>Memory Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:2]</td> </tr> </table> <p>This field contains the graphics memory address of the data on which atomic operation has to be performed. Atomic operation can be performed on data granularity of 4B, 8B or 16B and hence the Address has to be correspondingly aligned to 4B,8B or 16B respectively.</p>	Format:	GraphicsAddress[31:2]
	Format:	GraphicsAddress[31:2]		
1:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
2	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
15:0	<p>Memory Address High</p> <p>This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space.</p>			
3	31:0	<p>Operand1 Data Dword 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U32</td> </tr> </table> <p>Dword0 of Operand1 when Inline Data mode is set.</p>	Format:	U32
		Format:	U32	
<p>Operand2 Data Dword 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U32</td> </tr> </table> <p>Dword0 of Operand2 when Inline Data mode is set.</p>	Format:	U32		
Format:	U32			
5	31:0	<p>Operand1 Data Dword 1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U32</td> </tr> </table> <p>Dword1 of Operand1 when Inline Data mode is set.</p>	Format:	U32
		Format:	U32	
<p>Operand2 Data Dword 1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U32</td> </tr> </table> <p>Dword1 of Operand2 when Inline Data mode is set.</p>	Format:	U32		
Format:	U32			
7	31:0	<p>Operand1 Data Dword 2</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U32</td> </tr> </table> <p>Dword2 of Operand1 when Inline Data mode is set.</p>	Format:	U32
		Format:	U32	
<p>Operand2 Data Dword 2</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U32</td> </tr> </table> <p>Dword2 of Operand2 when Inline Data mode is set.</p>	Format:	U32		
Format:	U32			
9	31:0	<p>Operand1 Data Dword 3</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U32</td> </tr> </table> <p>Dword3 of Operand1 when Inline Data mode is set.</p>	Format:	U32
		Format:	U32	
<p>Operand2 Data Dword 3</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U32</td> </tr> </table> <p>Dword3 of Operand2 when Inline Data mode is set.</p>	Format:	U32		
Format:	U32			



MI_BATCH_BUFFER_END

MI_BATCH_BUFFER_END			
Source:	VideoEnhancementCS		
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	Bit	Description	
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
	22:0	Reserved	
		Format:	MBZ



MI_BATCH_BUFFER_END

MI_BATCH_BUFFER_END		
Source:	BlitterCS	
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	Bit	Description
0	31:29	Command Type Default Value: 0h MI_COMMAND
	28:23	MI Command Opcode Default Value: 0Ah MI_BATCH_BUFFER_END
	22:0	Reserved Format: MBZ



MI_BATCH_BUFFER_END

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	Bit	Description
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	
22:0	Reserved	
	Format: MBZ	



MI_BATCH_BUFFER_END

MI_BATCH_BUFFER_END			
Source:	VideoCS		
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	Bit	Description	
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
	22:0	Reserved	
		Format:	MBZ



MI_BATCH_BUFFER_START

MI_BATCH_BUFFER_START		
Source:	CommandStreamer	
Length Bias:	2	
<p>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 <i>MI Functions</i>. The batch buffer can be specified as privileged or non-privileged, 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 <i>MI Functions</i>.</p>		
Programming Notes		
<ul style="list-style-type: none"> 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. 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. 		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 0h MI_COMMAND
		Format: OpCode
	28:23	MI Command Opcode
		Default Value: 31h MI_BATCH_BUFFER_START
		Format: OpCode
22	Second Level Batch Buffer	
	<p>The command streamer contains three storage elements; one for the ring head address, one for the batch head address, and one for the second level batch head address. When performing batch buffer chaining, hardware simply updates the head pointer of the first level batch address storage. There is no stack in hardware. When this bit is set, hardware uses the 2nd level batch head address storage element. Upon MI_BATCH_BUFFER_END, it will automatically return to the first level batch buffer address. This allows hardware to mimic a simple 3-level stack.</p>	
	Value	Name
	Description	
0h	First level batch	Place the batch buffer address in the 1st (traditional) level batch address storage element.
1h	Second	Place the batch buffer address in the



MI_BATCH_BUFFER_START		
	level batch	second-level batch address storage element.
Programming Notes		
<p>Within a second level batch buffer there can't be any chained batch buffers. MI_BATCH_BUFFER_START command is not allowed inside a second level batch buffer.</p> <p>Setting this bit when the command is parsed in the ring, will cause the command to be ignored. This bit must only be set in a first level batch buffer.</p>		
21:20	Reserved	
	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS
	Format:	MBZ
21:20	Reserved	
	Source:	RenderCS
	Format:	MBZ
19	Reserved	
	Source:	RenderCS
	Format:	MBZ
18:17	Reserved	
	Format:	MBZ
16	Reserved	
	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS
	Format:	MBZ
16	Add Offset Enable	
	Source:	RenderCS
	Format:	Enable
<p>If this bit is set then the value stored in the BB_OFFSET MMIO register will be added to the Batch Buffer Start Address and the summation will be used as the address to fetch from memory.</p> <p>Specific to the render command stream only.</p>		
15	Reserved	
	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS
	Format:	MBZ



MI_BATCH_BUFFER_START					
15	<p>Predication Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Source:</td> <td>RenderCS</td> </tr> </table> <p>This bit is used to enable predication of this command. If this bit is set and Bit 0 of the MI_PREDICATE_RESULT_1 register is clear, this command is ignored. Otherwise the command is performed normally. Specific to the Render command stream only.</p>	Source:	RenderCS		
Source:	RenderCS				
14:11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ				
10	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Source:</td> <td>BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS	Format:	MBZ
Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS				
Format:	MBZ				
10	<p>Resource Streamer Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Source:</td> <td>RenderCS</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>When this bit is set, the Resource Streamer will execute the batch buffer. When this bit is clear the Resource Streamer will not execute the batch buffer. Specific to the Render command stream only.</p>	Source:	RenderCS	Format:	Enable
Source:	RenderCS				
Format:	Enable				
9	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Source:</td> <td>RenderCS, BlitterCS</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Source:	RenderCS, BlitterCS	Format:	MBZ
Source:	RenderCS, BlitterCS				
Format:	MBZ				
9	<p>Reserved</p>				
8	<p>Address Space Indicator</p> <p>Batch buffers accessed via PPGTT are considered as non-privileged. Certain operations (e.g., MI_STORE_DATA_IMM commands to GGTT memory) are prohibited within non-privileged buffers. More details mentioned in User Mode Privileged command section. When MI_BATCH_BUFFER_START command is executed from within a batch buffer (i.e., is a "chained" or "second level" batch buffer command), the current active batch buffer's "Address Space Indicator" and this field determine the "Address Space Indicator" of the next buffer in the chain.</p> <ul style="list-style-type: none"> • Chained or Second level batch buffer can be in GGTT or PPGTT if the parent batch buffer is in GGTT. • Chained or Second level batch buffer can only be in PPGTT if the parent batch buffer is in PPGTT. This is enforced by Hardware. 				



MI_BATCH_BUFFER_START											
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>GGTT</td> <td>This batch buffer is located in GGTT memory and is privileged.</td> </tr> <tr> <td>1h</td> <td>PPGTT</td> <td>This batch buffer is located in PPGTT memory and is Non-Privileged.</td> </tr> </tbody> </table>	Value	Name	Description	0h	GGTT	This batch buffer is located in GGTT memory and is privileged.	1h	PPGTT	This batch buffer is located in PPGTT memory and is Non-Privileged.
	Value	Name	Description								
	0h	GGTT	This batch buffer is located in GGTT memory and is privileged.								
	1h	PPGTT	This batch buffer is located in PPGTT memory and is Non-Privileged.								
	<table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">This field must be '0' unless the Per-Process GTT Enable is '1'</td> </tr> </tbody> </table>	Programming Notes		This field must be '0' unless the Per-Process GTT Enable is '1'							
Programming Notes											
This field must be '0' unless the Per-Process GTT Enable is '1'											
7:0	<p>DWord Length</p> <table border="1"> <tr> <td>Default Value:</td> <td>1h Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n</td> </tr> </table> <p>Total - Bias. Excludes DWord (0,1).</p>	Default Value:	1h Excludes DWord (0,1)	Format:	=n						
Default Value:	1h Excludes DWord (0,1)										
Format:	=n										
63:48	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>These bits are reserved for future expansion of the supported Virtual Address range, and are not to be used for other purposes.</p>	Format:	MBZ								
Format:	MBZ										
<p>1..2 GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and MBZ. Some GraphicsAddress fields only specify the upper address bits. For example, GraphicsAddress[47:12] is a 4KB page address.</p>	47:2	<p>Batch Buffer Start Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:2]</td> </tr> </table>	Format:	GraphicsAddress[47:2]							
	Format:	GraphicsAddress[47:2]									
	1:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ										



MI_CLFLUSH

MI_CLFLUSH				
Source:	RenderCS			
Length Bias:	2			
Flushes out the page given in the command out to system memory. This command is specific to the render engine and is not privileged.				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	0h MI_COMMAND	
		Format:	OpCode	
	28:23	MI Command Opcode		
		Default Value:	27h Store DW MI_CLFLUSH	
		Format:	OpCode	
	22	Use Global GTT		
		Value	Name	Description
		0h	Per Process Graphics Address	
		1h	Global Graphics Address	This command will use the global GTT to translate the Address.
21:10	Reserved			
	Format:	MBZ		
9:0	DWord Length			
	Format:	n Total Length - 2		
	Value	Name	Description	
	1h	[Default]	Excludes DWord (0,1)	
1..2	63:12	Page Base Address		
		Format:	GraphicsAddress63-12	
<p>4KB aligned Page Address for which software requires a hardware flush to DRAM. GraphicsAddress is 64-bit value [63:0], but only a portion of it is used by the hardware. The uppermost bits are ignored and MBZ.</p> <p>Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] is a 4KB page address.</p>				
11:6	11:6	Starting Cacheline Offset		
		Format:	U6	
Zero based starting cacheline offset in to the Page Base Address.				



MI_CLFLUSH		
	5:0	Reserved
		Format: MBZ
3..n	31:0	DW Representing a Half Cache Line
		Format: MBZ
		The information given to hardware is the DW itself, not the contents. Hardware uses the DW count of the command to determine the offset from the base to flush out. The offset is ½ cache line (8 DW = 1HW) granular so for a full page, the command will need 4096 bytes / 4 bytes per DW / 8 DW per HW = 128 DW.
		Programming Notes
		Always even number of "DW Representing 1/2 cacheline" terms must be programmed.



MI_CONDITIONAL_BATCH_BUFFER_END

MI_CONDITIONAL_BATCH_BUFFER_END			
Source:	CommandStreamer		
Length Bias:	2		
Description			
<p>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.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	36h MI_CONDITIONAL_BATCH_BUFFER_END
		Format:	OpCode
	22	Use Global GTT	
		Default Value:	0h
		Format:	Boolean
	<p>If set, this command will use the global GTT to translate the Compare Address and this command must be executing from a privileged (secure) batch buffer. If clear, the PPGTT will be used to translate the Compare Address.</p>		
21	Compare Semaphore		
	Default Value:	0h	
	Format:	Boolean	
<p>If set, the value from the Compare Data Dword is compared to the value from the Compare Address in memory. If the value at Compare Address is greater than the Compare Data Dword, execution of current command buffer should continue. If clear, the parser will continue to the next command and not exit the batch buffer.</p>			
20	Reserved		



MI_CONDITIONAL_BATCH_BUFFER_END															
	19	Compare Mask Mode <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Compare Mask Mode Disabled</td> <td>Compare address points to Dword in memory consisting of Data Dword(DW0). HW will compare Data Dword(DW0) against Semaphore Data Dword.</td> </tr> <tr> <td>1h</td> <td>Compare Mask Mode Enabled</td> <td>Compare address points to Qword in memory consisting of compare Mask (DW0) and Data Dword(DW1). HW will do AND operation on Mask(DW0) with Data Dword(DW1) and then compare the result against Semaphore Data Dword.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">When "Compare Mask Mode" is enabled, "Compare Address" must be qword aligned.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Compare Mask Mode Disabled	Compare address points to Dword in memory consisting of Data Dword(DW0). HW will compare Data Dword(DW0) against Semaphore Data Dword.	1h	Compare Mask Mode Enabled	Compare address points to Qword in memory consisting of compare Mask (DW0) and Data Dword(DW1). HW will do AND operation on Mask(DW0) with Data Dword(DW1) and then compare the result against Semaphore Data Dword.	Programming Notes		When "Compare Mask Mode" is enabled, "Compare Address" must be qword aligned.	
	Value	Name	Description												
	0h	Compare Mask Mode Disabled	Compare address points to Dword in memory consisting of Data Dword(DW0). HW will compare Data Dword(DW0) against Semaphore Data Dword.												
	1h	Compare Mask Mode Enabled	Compare address points to Qword in memory consisting of compare Mask (DW0) and Data Dword(DW1). HW will do AND operation on Mask(DW0) with Data Dword(DW1) and then compare the result against Semaphore Data Dword.												
	Programming Notes														
	When "Compare Mask Mode" is enabled, "Compare Address" must be qword aligned.														
	18	Reserved													
	17:15	Reserved													
	14:12	Reserved													
	11:8	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ											
Format:	MBZ														
7:0	DWord Length <table border="1"> <tr> <td>Default Value:</td> <td>2h Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table>	Default Value:	2h Excludes DWord (0,1)	Format:	=n Total Length - 2										
Default Value:	2h Excludes DWord (0,1)														
Format:	=n Total Length - 2														
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 Compare Address is greater than this dword, the execution of the command buffer should continue.														
2..3	63:48 Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ														
Qword address to fetch Data Qword from memory. This field specifies the 4GB aligned base address of Gfx 4GB	47:3 Compare Address <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:3]</td> </tr> </table>	Format:	GraphicsAddress[47:3]												
Format:	GraphicsAddress[47:3]														



MI_CONDITIONAL_BATCH_BUFFER_END		
virtual address space within the host's 64-bit virtual address space. GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. Upper reserved bits are ignored and MBZ. Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] is a 4KB page address.	2:0	Reserved
		Format: MBZ



MI_COPY_MEM_MEM

MI_COPY_MEM_MEM				
Source:	VideoCS			
Length Bias:	2			
<p>The MI_COPY_MEM_MEM command reads a DWord from memory and stores the value of that DWord to back to memory. The source and destination addresses are specified in the command. The command temporarily halts command execution.</p>				
Programming Notes				
<p>This command should not be used within a "non-privileged" batch buffer to access global virtual space; doing so will be treated as privilege access violation. Refer to the "User Mode Privilege Command" in MI_BATCH_BUFFER_START command section to learn more about HW behavior on encountering a privilege access violation.</p> <p>This command can be used within ring buffers and/or privilege batch buffers to access global virtual space.</p>				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	0h MI_COMMAND	
		Format:	OpCode	
	28:23	MI Command Opcode		
		Default Value:	2Eh MI_MEM_TO_MEM	
		Format:	OpCode	
	22	Use Global GTT Source		
		It is allowed for this bit to be set when executing this command from a privileged (secure) batch buffer or ring buffer. This bit must be clear when programmed from within a non-privileged batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear.		
		Value	Name	Description
		0h	Per Process Graphics Address	
1h	Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.		
21	Use Global GTT Destination			
	It is allowed for this bit to be set when executing this command from a privileged (secure) batch buffer or ring buffer. This bit must be clear when programmed from within a non-privileged batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear.			
	Value	Name	Description	
	0h	Per Process Graphics Address		



MI_COPY_MEM_MEM			
	1h	Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.
	20:8	Reserved	
		Format:	MBZ
	7:0	DWord Length	
		Default Value:	3 Excludes DWord (0,1)
		Format:	=n Total Length - 2
1..2	63:2	Destination Memory Address	
		Format:	GraphicsAddress63-2
		Surface Type: MMIO Register	
	1:0	Reserved	
		Format:	MBZ
3..4	63:2	Source Memory Address	
		Format:	GraphicsAddress63-2
		Surface Type: MMIO Register	
	1:0	Reserved	
		Format:	MBZ



MI_COPY_MEM_MEM

MI_COPY_MEM_MEM				
Source:	VideoEnhancementCS			
Length Bias:	2			
<p>The MI_COPY_MEM_MEM command reads a DWord from memory and stores the value of that DWord to back to memory. The source and destination addresses are specified in the command. The command temporarily halts command execution.</p>				
Programming Notes				
<p>This command should not be used within a "non-privileged" batch buffer to access global virtual space; doing so will be treated as privilege access violation. Refer to the "User Mode Privilege Command" in MI_BATCH_BUFFER_START command section to learn more about HW behavior on encountering a privilege access violation.</p> <p>This command can be used within ring buffers and/or privilege batch buffers to access global virtual space.</p>				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	0h MI_COMMAND	
		Format:	OpCode	
	28:23	MI Command Opcode		
		Default Value:	2Eh MI_MEM_TO_MEM	
		Format:	OpCode	
	22	Use Global GTT Source		
		It is allowed for this bit to be set when executing this command from a privileged (secure) batch buffer or ring buffer. This bit must be clear when programmed from within a non-privileged batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear.		
		Value	Name	Description
		0h	Per Process Graphics Address	
1h		Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.	
21	Use Global GTT Destination			
	It is allowed for this bit to be set when executing this command from a privileged (secure) batch buffer or ring buffer. This bit must be clear when programmed from within a non-privileged batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear.			
	Value	Name	Description	
	0h	Per Process Graphics Address		



MI_COPY_MEM_MEM			
	1h	Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.
	20:8	Reserved	
		Format:	MBZ
	7:0	DWord Length	
		Default Value:	3 Excludes DWord (0,1)
		Format:	=n Total Length - 2
1..2	63:2	Destination Memory Address	
		Format:	GraphicsAddress63-2
		Surface Type: MMIO Register	
	1:0	Reserved	
		Format:	MBZ
3..4	63:2	Source Memory Address	
		Format:	GraphicsAddress63-2
		Surface Type: MMIO Register	
	1:0	Reserved	
		Format:	MBZ



MI_COPY_MEM_MEM

MI_COPY_MEM_MEM			
Source:	BlitterCS		
Length Bias:	2		
<p>The MI_COPY_MEM_MEM command reads a DWord from memory and stores the value of that DWord to back to memory. The source and destination addresses are specified in the command. The command temporarily halts command execution.</p>			
Programming Notes			
<p>This command should not be used within a "non_privilege"batch buffer to access global virtual space, doing so will be treated as privilege access violation. Refer "User Mode Privilege Command" in MI_BATCH_BUFFER_START command section to know HW behavior on encountering privilege access violation. This command can be used within ring buffers and/or privilege batch buffers to access global virtual space.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	2Eh MI_COPY_MEM_MEM
		Format:	OpCode
	22	Use Global GTT Source	
		It is allowed for this bit to be set when executing this command from a privileged (secure) batch buffer or ring buffer. This bit must be clear when programmed from within a non-privileged batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear.	
		Value	Name
		Description	
0h	Per Process Graphics Address		
1h	Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.	



MI_COPY_MEM_MEM											
	21	<p>Use Global GTT Destination</p> <p>It is allowed for this bit to be set when executing this command from a privileged (secure) batch buffer or ring buffer. This bit must be clear when programmed from within a non-privileged batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Per Process Graphics Address</td> <td></td> </tr> <tr> <td>1h</td> <td>Global Graphics Address</td> <td>This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Per Process Graphics Address		1h	Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.
	Value	Name	Description								
	0h	Per Process Graphics Address									
	1h	Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.								
20:8	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
7:0	<p>DWord Length</p> <table border="1"> <tr> <td>Default Value:</td> <td>3 Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2</td> </tr> </table>	Default Value:	3 Excludes DWord (0,1)	Format:	=n Total Length - 2						
Default Value:	3 Excludes DWord (0,1)										
Format:	=n Total Length - 2										
<p>1..2</p> <p>Surface Type: MMIO Register GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and MBZ.</p>	63:48	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
47:0	<p>Destination Memory Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:0]</td> </tr> </table>	Format:	GraphicsAddress[47:0]								
Format:	GraphicsAddress[47:0]										
<p>3..4</p> <p>Surface Type: MMIO Register GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and MBZ.</p>	63:48	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
47:0	<p>Source Memory Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:0]</td> </tr> </table>	Format:	GraphicsAddress[47:0]								
Format:	GraphicsAddress[47:0]										



MI_COPY_MEM_MEM

MI_COPY_MEM_MEM		
Source:	RenderCS	
Length Bias:	2	
<p>The MI_COPY_MEM_MEM command reads a DWord from memory and stores the value of that DWord to back to memory. The source and destination addresses are specified in the command. The command temporarily halts command execution.</p>		
Programming Notes		
<p>This command should not be used within a "non_privilege"batch buffer to access global virtual space, doing so will be treated as privilege access violation. Refer "User Mode Privilege Command" in MI_BATCH_BUFFER_START command section to know HW behavior on encountering privilege access violation. This command can be used within ring buffers and/or privilege batch buffers to access global virtual space.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 0h MI_COMMAND
	Format: OpCode	
	28:23	MI Command Opcode
Default Value: 2Eh MI_COPY_MEM_TO_MEM		
Format: OpCode		
22	Use Global GTT Source	
	It is allowed for this bit to be set when executing this command from a privileged (secure) batch buffer or ring buffer. This bit must be clear when programmed from within a non-privileged batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear.	
	Value	Name
	0h	Per Process Graphics Address
1h	Global Graphics Address	It is allowed for this bit to be set when executing this command from a privileged (secure) batch buffer or ring buffer. This bit must be clear when programmed from within a non-privileged batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear.
21	Use Global GTT Destination	
	This bit will be ignored and treated as if clear when executing from a non-privileged batch buffer. It is allowed for this bit to be clear when executing this command from a privileged (secure) batch buffer. This bit <i>must</i> be '1' if the Per Process GTT Enable bit is clear. This bit will determine write to memory uses Global or Per Process GTT.	
	Value	Name
0h	Per Process Graphics Address	



MI_COPY_MEM_MEM			
	1h	Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.
	20:8	Reserved	
		Format:	MBZ
	7:0	DWord Length	
		Default Value:	3 Excludes DWord (0,1)
		Format:	=n Total Length - 2
1..2	63:2	Destination Memory Address	
		Format:	GraphicsAddress[63:2]
		Surface Type: MMIO Register This field specifies the address of the memory location where the value fetched specified in the DWord address above will be written. The address specifies the DWord location of the data. Range = GraphicsVirtualAddress[63:2] for a DWord register GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].	
	1:0	Reserved	
		Format:	MBZ
3..4	63:2	Source Memory Address	
		Format:	GraphicsAddress[63:2]
		Surface Type: MMIO Register This field specifies the address of the memory location where the value is located that will be written to the address below. The address specifies the DWord location of the data. Range = GraphicsVirtualAddress[63:2] for a DWord register GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].	
	1:0	Reserved	
		Format:	MBZ



MI_DISPLAY_FLIP

MI_DISPLAY_FLIP	
Source:	RenderCS, BlitterCS
Length Bias:	2
<p>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.</p> <p>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 (possibly on a future frame), or asynchronously (as soon as possible) to minimize rendering stalls at the cost of tearing artifacts.</p>	
Programming Notes	
<ol style="list-style-type: none"> 1. 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 command does not guarantee that outstanding flip operations have completed). The MI_WAIT_FOR_EVENT command can be used to provide this synchronization - 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. 2. After a display flip operation is requested, software is responsible for initiating any required synchronization with subsequent buffer clear or rendering operations. For multi-buffering (e.g., double buffering) operations, this will typically require updating SURFACE_STATE or the binding table to change the rendering (back) buffer. In addition, prior to any subsequent clear or rendering operations, software must typically ensure that the new rendering 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. 3. 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. 4. 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. <ul style="list-style-type: none"> • 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. 5. DWord 3 (Left Eye Display Buffer Base Address) must not be set with synchronous flips or asynchronous flips. It is only allowed to be sent with stereo 3D flips. 	



MI_DISPLAY_FLIP

Asynchronous flip completion time depends greatly on how much data has been prefetched for power savings, and can take up to 1 full frame to complete. For faster flip completion, disable FBC and render compression and allocate a small amount of data buffer for the plane.

"Command Streamer Plane Number" mapping to "Display Plane Name" are listed in dsisplay B-spec -"Plane capability and Interoperability".

DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	0h MI_COMMAND	
		Format:	OpCode	
	28:23		MI Command Opcode	
			Default Value:	14h MI_DISPLAY_FLIP
			Format:	OpCode
	22		Async Flip Indicator	
			Format:	Enable
	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 render pipe while DW2 is used by the display hardware.			
	21:19		Reserved	
			Format:	MBZ
	18:17		Reserved	
			Format:	MBZ
16:14		Reserved		
		Format:	MBZ	
13		Reserved		
		Format:	MBZ	
12:8		Display Plane Select		
		Value	Name	
		0h	Display Plane 1	
		1h	Display Plane 2	
		2h	Display Plane 3	
		3h	Reserved	
		4h	Display Plane 4	
		5h	Display Plane 5	
		6h	Display Plane 6	
		7h	Display Plane 7	
		8h	Display Plane 8	
9h	Display Plane 9			



MI_DISPLAY_FLIP													
		<table border="1"> <tr> <td>Ah</td> <td>Display Plane 10</td> </tr> <tr> <td>Bh</td> <td>Display Plane 11</td> </tr> <tr> <td>Ch</td> <td>Display Plane 12</td> </tr> <tr> <td>[Dh-1Fh]</td> <td>Reserved</td> </tr> </table>	Ah	Display Plane 10	Bh	Display Plane 11	Ch	Display Plane 12	[Dh-1Fh]	Reserved			
Ah	Display Plane 10												
Bh	Display Plane 11												
Ch	Display Plane 12												
[Dh-1Fh]	Reserved												
	7:0	<p>DWord Length</p> <table border="1"> <tr> <td>Format:</td> <td>=n</td> </tr> </table> <p>Total Length - 2. Excludes DWord (0,1). For Synchronous Flips and Asynchronous Flips, this field must be programmed to 1h for a total length of 3. For Stereo 3D Flips, this field must be programmed to 2h for a total length of 4.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Exists If</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td>[Default]</td> <td>(([Flip Type]!='Stereo 3D Flip')</td> </tr> <tr> <td>2h</td> <td>[Default]</td> <td>(([Flip Type]='Stereo 3D Flip')</td> </tr> </tbody> </table>	Format:	=n	Value	Name	Exists If	1h	[Default]	(([Flip Type]!='Stereo 3D Flip')	2h	[Default]	(([Flip Type]='Stereo 3D Flip')
Format:	=n												
Value	Name	Exists If											
1h	[Default]	(([Flip Type]!='Stereo 3D Flip')											
2h	[Default]	(([Flip Type]='Stereo 3D Flip')											
1	31	<p>Stereoscopic 3D Mode</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit must be set if the Extra Display Buffer Address is part of this command. This bit is used to notify the display there is an extra DW before processing the Display Flip.</p>	Default Value:	0h	Format:	Enable							
Default Value:	0h												
Format:	Enable												
	30:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												
	15:6	<p>Display Buffer Pitch</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> <p><i>For Synchronous Flips and Stereo 3D Flips only</i>, this field specifies the pitch of the new display buffer. For Asynchronous Flips, this parameter is programmed so that all the flips in a flip chain should maintain the same pitch as programmed with the last synchronous flip or stereo 3D flip or direct through MMIO. See the Display Plane Stride register for details.</p>	Default Value:	0h	Format:	U10							
Default Value:	0h												
Format:	U10												
	5:3	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ												



MI_DISPLAY_FLIP			
2:0	Tile Parameter		
	Format:	Enable	
	For Asynchronous Flips, this parameter cannot be changed. All the flips in a flip chain should maintain the same tile parameter as programmed with the last synchronous flip or direct through MMIO.		
	Value	Name	Description
	0h	Linear [Default]	For Synchronous Flips Only
	1h	Tiled X	
	2h-3h	Reserved	
2	31:12	Display Buffer Base Address	
		Format:	GraphicsAddress[31:12]
		This field specifies Bits 31:12 of the Graphics Address of the new display buffer. In stereo 3D mode this is the right eye base address. In non-stereo 3D mode this is the only base address. (Refer to the Display Address Start Address Register description in the Display Registers chapter).	
		Programming Notes	
		<ul style="list-style-type: none"> The Display buffer must reside completely in Main Memory. This address is always translated via the global (rather than per-process) GTT. 	
11:3	Reserved		
	Format:	MBZ	
Reserved			
1:0	Flip Type		
	This field specifies whether the flip operation should be performed asynchronously to vertical retrace.		
	Value	Name	Description
	00b	Sync Flip [Default]	The flip will occur during the vertical blanking interval - thus avoiding any tearing artifacts.
	01b	Async Flip	The flip will occur "as soon as possible" - and may exhibit tearing artifacts
	10b	Stereo 3D Flip	The flip will occur during the vertical blanking interval (left or right eye blank selectable through display MMIO register) - thus avoiding any tearing artifacts.
11b	Reserved		
Programming Notes			
<ul style="list-style-type: none"> The Display Buffer Pitch and Tile parameter cannot be changed for asynchronous flips 			



MI_DISPLAY_FLIP						
		<p>(i.e., the new buffer must have the same pitch/tile format as the previous buffer).</p> <ul style="list-style-type: none"> • For Asynch Flips the Buffers used must be 32KB aligned. • Asynch flips are supported on primary or universal planes only. <hr/> <ul style="list-style-type: none"> • For Stereo 3D flips, both left and right eye buffers must have the same pitch and tile format. 				
3	31:12	<p>Left Eye Display Buffer Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>This field specifies Bits 31:12 of the Graphics Address of the new display buffer for the stereo 3D left eye. In non-stereo 3D mode this address is not used. (Refer to the Display Address Start Address Register description in the Display Registers chapter).</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table> <ul style="list-style-type: none"> • The Display buffer must reside completely in Main Memory. • This address is always translated via the global (rather than per-process) GTT. 	Format:	GraphicsAddress[31:12]	Programming Notes	
	Format:	GraphicsAddress[31:12]				
Programming Notes						
11:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					



MI_FLUSH_DW

MI_FLUSH_DW		
Source:	VideoEnhancementCS	
Length Bias:	2	
<p>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:</p> <ul style="list-style-type: none"> • Flush any dirty data to memory. • Invalidate the TLB cache inside the hardware <p>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).</p>		
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
	21	Store Data Index Format: U1 <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center; margin: 0;">Description</p> <p>Ring Buffer Mode Scheduling: 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 global hardware status page. This bit only applies to the Global HW status page. If this field is 1, the Destination Address Type in this command must be set to 1 (GGTT).</p> <p>Execlist Mode Scheduling: This field is valid only if the post-sync operation is not 0. If this bit is set, the store data address is index into the global hardware status page when destination address type in the command is set to 1 (GGTT). The store data address is index into the per-process hardware status page when destination address type in the command is set to 0 (PPGTT).</p> </div>
	20:19	Reserved Format: MBZ
	18	TLB Invalidate Format: U1 <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center; margin: 0;">Description</p> <p>If ENABLED, all TLBs belonging to Video Enhancement Engine will be invalidated once the flush</p> </div>



MI_FLUSH_DW																	
	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	Reserved Format: _____ MBZ																
16	Reserved Format: _____ MBZ																
15:14	Post-Sync Operation <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No Write</td> <td>No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc.</td> </tr> <tr> <td>1h</td> <td>Write Immediate Data</td> <td>Write the QWord containing Immediate Data Low, High DWs to the Destination Address</td> </tr> <tr> <td>2h</td> <td>Reserved</td> <td>Reserved</td> </tr> <tr> <td>3h</td> <td>Write TIMESTAMP register</td> <td>Write the TIMESTAMP register to the Destination Address. The upper 28 bits of the TIMESTAMP register are tied to '0'.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> If executed in non-secure batch buffer, the address given will be in a PPGTT address space. If in a secure ring or batch, address given will be in GGTT space		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 Data	Write the QWord containing Immediate Data Low, High DWs to the Destination Address	2h	Reserved	Reserved	3h	Write TIMESTAMP register	Write the TIMESTAMP register to the Destination Address. The upper 28 bits of the TIMESTAMP register are tied to '0'.
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 Data	Write the QWord containing Immediate Data Low, High DWs to the Destination Address															
2h	Reserved	Reserved															
3h	Write TIMESTAMP register	Write the TIMESTAMP register to the Destination Address. The upper 28 bits of the TIMESTAMP register are tied to '0'.															
13:10	Reserved Format: _____ MBZ																
9	Flush LLC Format: _____ Enable If enabled, at the end of the current MI_FLUSH_DW the last level cache is cleared of all the cachelines which have been determined as being part of the Frame Buffer.																
8	Notify Enable Format: _____ U1 If ENABLED, a Sync Completion Interrupt will be generated (if enabled by the MI Interrupt Control registers) once the sync operation is complete. See Interrupt Control Registers in Memory Interface Registers for details.																
7	Reserved Format: _____ MBZ																
6	Reserved Format: _____ MBZ																



MI_FLUSH_DW											
	5:0	DWord Length									
		Format: <table border="1" style="display: inline-table;"><tr><td>=n Total Length - 2</td></tr></table>	=n Total Length - 2								
		=n Total Length - 2									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>3h</td> <td>Excludes DWord (0,1) = 2 for DWord, 3 for QWord [Default]</td> </tr> </tbody> </table>	Value	Name	3h	Excludes DWord (0,1) = 2 for DWord, 3 for QWord [Default]							
Value	Name										
3h	Excludes DWord (0,1) = 2 for DWord, 3 for QWord [Default]										
1	31:3	Address									
		Format: <table border="1" style="display: inline-table;"><tr><td>GraphicsAddress[31:3]U28</td></tr></table> This field specifies Bits 31:3 of the Address where the DWord or QWord will be stored. Note that the address can only be QWord aligned, irrespective of data size.	GraphicsAddress[31:3]U28								
	GraphicsAddress[31:3]U28										
	2	Destination Address Type									
Defines address space of Destination Address											
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>PPGTT</td> <td>Use PPGTT address space for DW write</td> </tr> <tr> <td>1h</td> <td>GGTT</td> <td>Use GGTT address space for DW write</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Ignored if "No write" is the selected in Operation.</td> </tr> </tbody> </table>		Value	Name	Description	0h	PPGTT	Use PPGTT address space for DW write	1h	GGTT	Use GGTT address space for DW write	Programming Notes
Value	Name	Description									
0h	PPGTT	Use PPGTT address space for DW write									
1h	GGTT	Use GGTT address space for DW write									
Programming Notes											
Ignored if "No write" is the selected in Operation.											
1:0	Reserved										
	Format: <table border="1" style="display: inline-table;"><tr><td>MBZ</td></tr></table>	MBZ									
MBZ											
2	31:16	Reserved									
		Format: <table border="1" style="display: inline-table;"><tr><td>MBZ</td></tr></table>	MBZ								
MBZ											
	15:0	Address High									
		Format: <table border="1" style="display: inline-table;"><tr><td>GraphicsAddress[47:32]U64</td></tr></table> This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space	GraphicsAddress[47:32]U64								
GraphicsAddress[47:32]U64											
3..4	31:0	Immediate Data									
		<p>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</p> <p>To avoid hitting a known hardware bug, drivers cannot send a QW write when bit 5 of the address is '1'</p>									



MI_FLUSH_DW

MI_FLUSH_DW		
Source:	BlitterCS	
Length Bias:	2	
<p>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</p> <p>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).</p>		
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: U1
	21	Store Data Index
		Format: U1
<p style="text-align: center;">Description</p> <p>Ring Buffer Mode Scheduling: 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 global hardware status page. This bit only applies to the Global HW status page. If this field is 1, the Destination Address Type in this command must be set to 1 (GGTT).</p> <p>Execlist Mode Scheduling: This field is valid only if the post-sync operation is not 0. If this bit is set, the store data address is index into the global hardware status page when destination address type in the command is set to 1 (GGTT). The store data address is index into the per-process hardware status page when destination address type in the command is set to 0 (PPGTT).</p>		
20:19	Reserved	
	Format: MBZ	
18	TLB Invalidate	
	Format: U1	
	<p style="text-align: center;">Description</p> <p>If ENABLED, all TLBs belonging to Blitter 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.</p>	



MI_FLUSH_DW		
	If GFX_MODE (0x229c) bit 13, this command will cause a config write to MMIO register space with the address 0x4f100.	
17	Reserved	
	Format:	MBZ
16	Reserved	
	Format:	MBZ
15:14	Post-Sync Operation	
	BitFieldDesc	
	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 Data QWord Write the QWord containing Immediate Data Low, High DWs to the Destination Address
	2h	Reserved Reserved
	3h	Write TIMESTAMP Register Write the TIMESTAMP register to the Destination Address. The upper 28 bits of the TIMESTAMP register are tied to '0'.
	Programming Notes	
	If executed in a PPGTT (non-secure) batch buffer, the post-sync op is always inhibited. This command does not write anything out to memory. [For all other devices]: If executed in a non-secure batch buffer, the address given is in a PPGTT address space. If in a secure ring or batch, the address given is in GGTT space.	
13:10	Reserved	
	Format:	MBZ
9	Flush LLC	
	Format:	Enable
	If enabled, at the end of the current MI_FLUSH_DW the last level cache is cleared of all the cachelines which have been determined as being part of the Frame Buffer.	
8	Notify Enable	
	Format:	U1
	If ENABLED, a Sync Completion Interrupt will be generated (if enabled by the MI Interrupt Control registers) once the sync operation is complete. See Interrupt Control Registers in Memory Interface Registers for details.	
7:6	Reserved	
	Format:	MBZ
5:0	DWord Length	
	Format:	=n Total Length - 2



		MI_FLUSH_DW	
		Value	Name
		3h	Excludes DWord (0,1) = 2 for DWord, 3 for QWord [Default]
1..2	63:48	Reserved	
		Format:	MBZ
	47:2	Destination Address	
		Format:	GraphicsAddress[47:2]
	1:0	Reserved	
		Format:	MBZ
3..4	63:0	Immediate Data	
		This field specifies the DWord value to be written to the targeted location. Only valid when 15:14 in header is set to 1h.	
		Programming Notes	
		To avoid hitting a known hardware bug, drivers cannot send a QW write when bit 5 of the address is '1'	



MI_FLUSH_DW

MI_FLUSH_DW		
Source:	VideoCS	
Length Bias:	2	
<p>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).</p>		
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
	21	Store Data Index Format: U1 <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Description</div> <p>Ring Buffer Mode Scheduling: 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 global hardware status page. This bit only applies to the Global HW status page. If this field is 1, the Destination Address Type in this command must be set to 1 (GGTT).</p> <p>Execlist Mode Scheduling: This field is valid only if the post-sync operation is not 0. If this bit is set, the store data address is index into the global hardware status page when destination address type in the command is set to 1 (GGTT). The store data address is index into the per-process hardware status page when destination address type in the command is set to 0 (PPGTT).</p>
	20:19	Reserved Format: MBZ
	18	TLB Invalidate Format: U1 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.
	17	Reserved Format: MBZ
	16	Reserved Format: MBZ



MI_FLUSH_DW		
15:14	Post-Sync Operation BitFieldDesc	
	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 Data	HW 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	Reserved
3h		Write the TIMESTAMP register to the Destination Address. The upper 28 bits of the TIMESTAMP register are tied to '0'.
Programming Notes		
If executed in a PPGTT (non-secure) batch buffer, the post-sync op is always inhibited. This command does not write anything to memory.		
13:10	Reserved	
	Format:	MBZ
9	Flush LLC	
	Format:	Enable
If enabled, at the end of the current MI_FLUSH_DW the last level cache is cleared of all the cachelines which have been determined as being part of the Frame Buffer.		
8	Notify Enable	
	Format:	U1
If ENABLED, a Sync Completion Interrupt will be generated (if enabled by the MI Interrupt Control registers) once the sync operation is complete. See Interrupt Control Registers in Memory Interface Registers for details.		
7	Video Pipeline Cache invalidate	
	Format:	U1
Enable the invalidation of the video cache at the end of this flush		
6	Reserved	
	Format:	MBZ
5:0	DWord Length	
	Format:	=n Total Length - 2



MI_FLUSH_DW												
		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">3h</td> <td>Excludes DWord (0,1) = 2 for DWord, 3 for QWord [Default]</td> </tr> </tbody> </table>	Value	Name	3h	Excludes DWord (0,1) = 2 for DWord, 3 for QWord [Default]						
Value	Name											
3h	Excludes DWord (0,1) = 2 for DWord, 3 for QWord [Default]											
1..2	63:48	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	47:3	Address <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:3]U64</td> </tr> </table> <p>This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space.</p> <p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by the hardware. Upper bits [63:48] are ignored and MBZ. Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] is a 4KB page address.</p>	Format:	GraphicsAddress[47:3]U64								
	Format:	GraphicsAddress[47:3]U64										
2	Destination Address Type Defines address space of Destination Address <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>PPGTT</td> <td>Use PPGTT address space for DW write</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>GGTT</td> <td>Use GGTT address space for DW write</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>Ignored if "No write" is the selected in Operation.</td> </tr> </tbody> </table>	Value	Name	Description	0h	PPGTT	Use PPGTT address space for DW write	1h	GGTT	Use GGTT address space for DW write	Programming Notes	Ignored if "No write" is the selected in Operation.
Value	Name	Description										
0h	PPGTT	Use PPGTT address space for DW write										
1h	GGTT	Use GGTT address space for DW write										
Programming Notes												
Ignored if "No write" is the selected in Operation.												
1:0	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
3..4	31:0	Immediate Data <table border="1"> <tr> <td>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</td> </tr> <tr> <td>To avoid hitting a known hardware bug, drivers cannot send a QW write when bit 5 of the address is '1'</td> </tr> </table>	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 send a QW write when bit 5 of the address is '1'								
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 send a QW write when bit 5 of the address is '1'												



MI_FORCE_WAKEUP

MI_FORCE_WAKEUP			
Source:	CommandStreamer		
Length Bias:	2		
<p>This command is used to communicate Force Wakeup request to PM unit. No functionality is performed by this command when none of the mask bits are set and is equivalent to NOOP. Example for usage model: VCS Ring Buffer: MI_FORCE_WAKEUP (Force Render Awake set to '1') MI_SEMPAHORE_SIGNAL (Signal context id 0xABC to Render Command Streamer) MI_FORCE_WAKEUP (Force Render Awake set to '0') MI_BATCH_BUFFER_START STATE Commands MI_FORCE_WAKEUP (Force Render Awake set to '1') MI_LOAD_REGISTER_IMMEDIATE (Load register 0x23XX in render command streamer with data 0xFF) MI_FORCE_WAKEUP (Force Render Awake set to '0') MI_BATCH_BUFFER_END</p>			
Programming Notes			
<p>This command must not be programmed in the render engine command streamer. Use PIPELINE_SELECT command to wake up media engines. These commands must be used in strict set then clear order. Do not send 2 wakes in a row or 2 clears in a row.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	1Dh MI_FORCE_WAKEUP
		Format:	OpCode
22:8	Reserved		
	Format:	MBZ	
7:0	DWord Length	Default Value:	0h
		Format:	=n
		Total Length - 2. Excludes DWord (0,1).	
1	31:16	Mask Bits	
		Format:	Mask[15:0]
		Programming Notes	
	Must be set to modify corresponding Bits 15:0. (Mask bits must not be set for reserved bits).		
	15:10	Reserved	
	Format:	MBZ	



MI_FORCE_WAKEUP							
9:8	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
7:5	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
4:2	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
1	<p>Force Render Awake</p> <table border="1"> <tr> <td>Source:</td> <td>BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>When set, Command Streamer sends message to PM to force awake render engine (next instructions require render engine awake). Command streamer waits for acknowledge from PM before parsing the next command.</p> <p>When reset, command streamer sends message to PM to disable force awake of render engine (next instructions do not require the render engine to be awake. Command streamer waits for acknowledge from PM before parsing the next command.</p> <table border="1" style="background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table> <p>Mask bit [17] has to be set for HW to look at this field when MI_FORCE_WAKEUP command is parsed.</p>	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS	Format:	U1	Programming Notes	
Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS						
Format:	U1						
Programming Notes							
1:0	<p>Reserved</p> <table border="1"> <tr> <td>Source:</td> <td>RenderCS</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Source:	RenderCS	Format:	MBZ		
Source:	RenderCS						
Format:	MBZ						
0	<p>Force Media Awake</p> <table border="1"> <tr> <td>Source:</td> <td>BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>When set, Command Streamer sends message to PM to force awake media engine (next instructions require media engine awake). Command streamer waits for acknowledge from PM before parsing the next command.</p> <p>When reset, command streamer sends message to PM to disable force awake of media engine (next instructions do not require the media engine to be awake). Command streamer waits for acknowledge from PM before parsing the next command.</p> <table border="1" style="background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table> <p>Mask bit [16] has to be set for HW to look at this field when MI_FORCE_WAKEUP command is parsed.</p>	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS	Format:	U1	Programming Notes	
Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS						
Format:	U1						
Programming Notes							



MI_LOAD_REGISTER_IMM

MI_LOAD_REGISTER_IMM						
Source:	CommandStreamer					
Length Bias:	2					
<p>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).</p> <p>Any offset that is to a destination outside of the GT core will allow the parser to continue once the cycle is at the GT boundry and not destination. Any other address will ensure the destination is updated prior to parsing the next command</p>						
Programming Notes		Source				
<p>Many MMIO bits require the engine to be IDLE prior to updating the value. Command streamer does not implicitly put in a pipeline flush in the cases a MMIO bit requires the engine to be IDLE. In the case there was a 3DPRIMITIVE command or GPGPU_WALKER command without any stalling PIPE_CONTROL, one must be inserted prior to a MI_LOAD_REGISTER_IMMEDIATE that is updating a bit that requires the engine to be IDLE.</p>		RenderCS				
<p>When executed from a non-privileged batch buffer, MMIO writes are only allowed to the registers listed in User Mode Non-Privileged Registers for the corresponding engine, any writes targeting the register not listed in the User Mode Non-Privileged Register will convert this command to a NOOP.</p>						
<p>The following addresses should NOT be used for LRIs:</p> <ol style="list-style-type: none"> 1. 0x8800 - 0x88FF 2. >= 0xC0000 <p>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.</p>						
<p>Programming an MMIO register is equivalent to programming a non-pipeline state to the hardware and hence an explicit stalling flush needs to be programmed prior to programming this command. However for certain MMIO registers based on their functionality doing an explicit stalling flush is exempted. Listed below are the exempted registers.</p> <ul style="list-style-type: none"> • 3DPRIM_END_OFFSET - Auto Draw End Offset • 3DPRIM_START_VERTEX - Load Indirect Start Vertex • 3DPRIM_VERTEX_COUNT - Load Indirect Vertex Count • 3DPRIM_INSTANCE_COUNT - Load Indirect Instance Count • 3DPRIM_START_INSTANCE - Load Indirect Start Instance • 3DPRIM_BASE_VERTEX - Load Indirect Base Vertex 		RenderCS				
DWord	Bit	Description				
0	31:29	<p>Command Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>0h MI_COMMAND</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	0h MI_COMMAND	Format:	OpCode
Default Value:	0h MI_COMMAND					
Format:	OpCode					



MI_LOAD_REGISTER_IMM						
	28:23	MI Command Opcode <table border="1"> <tr> <td>Default Value:</td> <td>22h MI_LOAD_REGISTER_IMM</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	22h MI_LOAD_REGISTER_IMM	Format:	OpCode
	Default Value:	22h MI_LOAD_REGISTER_IMM				
	Format:	OpCode				
	22:20	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	19	Reserved				
	18:13	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					
12	Reserved					
11:8	Byte Write Disables <table border="1"> <tr> <td>Format:</td> <td>Enable[4] Bit 8 corresponds to Data DWord [7:0]</td> </tr> </table> <p>Range: Must specify a valid register write operation If [11:8] is '1111b', then this command will behave as a NOOP. Otherwise, the value is forwarded to the destination register.</p>	Format:	Enable[4] Bit 8 corresponds to Data DWord [7:0]			
Format:	Enable[4] Bit 8 corresponds to Data DWord [7:0]					
7:0	DWord Length <table border="1"> <tr> <td>Default Value:</td> <td>1h Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2. Excludes DWord (0,1).</td> </tr> </table>	Default Value:	1h Excludes DWord (0,1)	Format:	=n Total Length - 2. Excludes DWord (0,1).	
Default Value:	1h Excludes DWord (0,1)					
Format:	=n Total Length - 2. Excludes DWord (0,1).					
1	31:23	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	22:2	Register Offset <table border="1"> <tr> <td>Format:</td> <td>MmioAddress[22:2]</td> </tr> </table> <p>This field specifies bits [22:2] of the offset into the Memory Mapped Register Range (i.e., this field specifies a DWord offset).</p>	Format:	MmioAddress[22:2]		
Format:	MmioAddress[22:2]					
1:0	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
2	31:0	Data DWord <table border="1"> <tr> <td>Mask:</td> <td>Bytes Write Disables</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>This field specifies the DWord value to be written to the targeted location.</p>	Mask:	Bytes Write Disables	Format:	U32
		Mask:	Bytes Write Disables			
		Format:	U32			



MI_LOAD_REGISTER_MEM

MI_LOAD_REGISTER_MEM			
Source:	RenderCS, BlitterCS, VideoCS, VideoEnhancementCS		
Length Bias:	2		
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 MMIO writes: <ul style="list-style-type: none"> • 0x8800 - 0x88FF • >= 0xC0000 Limited MMIO writes 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 MMIO write.			
This command should not be used within a non-privilege batch buffer to access global virtual space, doing so will be treated as privilege access violation. Refer "User Mode Privilege Command" in MI_BATCH_BUFFER_START command section to know HW behavior on encountering privilege access violation.			
This command is not allowed to update the privilege register range when executed from a non-privilege batch buffer.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
Default Value:		29h MI_LOAD_REGISTER_MEM	
22		Use Global GTT	
		Format:	Boolean
This bit if set when executing from a non-privileged batch buffer will be treated as privilege access violation. It is allowed for this bit to be clear when executing this command from a privileged (secure) batch buffer or ring buffer. This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.			
21		Async Mode Enable	
		Format:	Enable
If this bit is set then the command stream will not wait for completion of this command before executing the next command. Please refer to the LOAD_INDIRECT and Predicate registers for usage of this bit.			



MI_LOAD_REGISTER_MEM					
	20	Reserved			
	19	Reserved			
	18:8	Reserved			
	Format: MBZ				
	7:0	DWord Length			
Format: =n Total Length - 2. Excludes DWord (0,1).					
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>2h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>		Value	Name	2h	Excludes DWord (0,1) [Default]
Value	Name				
2h	Excludes DWord (0,1) [Default]				
1	31:23	Reserved			
	Format: MBZ				
	22:2	Register Address			
Format: MMIOAddress[22:2]					
This field specifies Bits 22:2 of the Register offset the DWord will be written to. As the register address must be DWord-aligned, Bits 1:0 of that address MBZ.					
1	1:0	Reserved			
	Format: MBZ				
2..3	63:2	Memory Address			
	Format: GraphicsAddress[63:2]				
This field specifies the address of the memory location where the register value specified in the DWord above will read from. The address specifies the DWord location of the data. Range = GraphicsVirtualAddress[63:2] for a DWord register GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].					
2..3	1:0	Reserved			
	Format: MBZ				



MI_LOAD_REGISTER_REG

MI_LOAD_REGISTER_REG		
Source:	CommandStreamer	
Length Bias:	2	
<p>The MI_LOAD_REGISTER_REG command reads from a source register location and writes that value to a destination register location.</p> <p>Any offset that is to a destination outside of the GT core will allow the parser to continue once the cycle is at the GT boundry and not destination. Any other address will ensure the destination is updated prior to parsing the next command</p>		
Programming Notes		
<p>The command temporarily halts commands that will cause cycles down the 3D pipeline.</p> <p>Destination register with mask implemented will not get updated unless the value read from source register has the bits corresponding to the mask bits set. Note that any mask implemented register when read returns "0" for the bits corresponding to mask location. When the source and destination are mask implemented registers, destination register will not get updated with the source register contents.</p> <p>This command is not allowed to update the privilege register range when executed from a non-privilege batch buffer.</p>		
Workaround		
<p>Workaround: Reads to MMIO registers outside GT (Source Register Address > 0x40000) are not supported.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value:
	Format:	OpCode
	28:23	MI Command Opcode
		Default Value:
	Format:	OpCode
	22:20	Reserved
		Format:
	19:18	Reserved
		Format:
	17:8	Reserved
		Format:
7:0	DWord Length	
	Default Value:	1h
Format:	=n Total Length - 2. Excludes DWord (0,1).	



MI_LOAD_REGISTER_REG		
1	31:23	Reserved Format: MBZ
	22:2	Source Register Address Format: MMIOAddress[22:2]MMIO_Register This field specifies Bits 22:2 of the Register offset the DWord will be written to. As the register address must be DWord-aligned, Bits 1:0 of that address MBZ.
	1:0	Reserved Format: MBZ
2	31:23	Reserved Format: MBZ
	22:2	Destination Register Address Format: MMIOAddress[22:2]MMIO_Register This field specifies Bits 22:2 of the Register offset the DWord will be written to. As the register address must be DWord-aligned, Bits 1:0 of that address MBZ.
	1:0	Reserved Format: MBZ



MI_LOAD_SCAN_LINES_EXCL

MI_LOAD_SCAN_LINES_EXCL			
Source:	BlitterCS		
Length Bias:	2		
<p>The MI_LOAD_SCAN_LINES_EXCL command is used to initialize the Scan Line Window registers for a specific Display Pipe. If the display refresh is <i>outside</i> this window the Display Engine asserts a signal that is used by the command parser to process the WAIT_FOR_EVENT command (i.e., the parser will wait while outside). This command overrides the Scan Line Window defined by any previous MI_LOAD_SCAN_LINES_INCL or MI_LOAD_SCAN_LINES_EXCL commands targeting the specific display pipe.</p> <p>Note: The two scan-line numbers are inclusive. If programmed to the same values, that single line defines the region in question.</p> <p>Always place an even number of MI_LOAD_SCAN_LINES_EXCL/INCL at a time into the ring buffer. If only a single MI_LOAD_SCAN_LINES_EXCL/INCL is desired, just add a second identical MI_LOAD_SCAN_LINES_EXCL/INCL command.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	13h MI_LOAD_SCAN_LINES_EXCL
		Format:	OpCode
	22	Reserved	
		Format:	MBZ
	21:19	Display Pipe Select	
		Format:	U3
		This field selects which Display Engine (pipe) this command is targeting.	
		Value	Name
0h		Display Pipe A	
1h		Display Pipe B	
2h, 3h		Reserved	
4h		Display Pipe C	
5h		Reserved	
6h, 7h		Reserved	
18:17	Reserved		
16:6	Reserved		
	Format:	MBZ	



MI_LOAD_SCAN_LINES_EXCL		
	5:0	DWord Length Default Value: 0h Excludes DWord (0,1) Format: =n Total Length - 2
1	31:16	Start Scan Line Number Format: U16 In scan lines, where scan line 0 is the first line of the display frame. This field specifies the starting scan line number of the Scan Line Window. Range: [0, Display Buffer height in lines-1]
	15:0	End Scan Line Number Format: U16 In scan lines, where scan line 0 is the first line of the display frame. This field specifies the ending scan line number of the Scan Line Window. Range: [0, Display Buffer height in lines-1]



MI_LOAD_SCAN_LINES_EXCL

MI_LOAD_SCAN_LINES_EXCL			
Source:	RenderCS		
Length Bias:	2		
<p>The MI_LOAD_SCAN_LINES_EXCL command is used to initialize the Scan Line Window registers for a specific Display Pipe. If the display refresh is outside this window the Display Engine asserts a signal that is used by the command parser to process the WAIT_FOR_EVENT command (i.e., the parser will wait while outside). This command overrides the Scan Line Window defined by any previous MI_LOAD_SCAN_LINES_INCL or MI_LOAD_SCAN_LINES_EXCL commands targeting the specific display pipe.</p> <p>Note: The two scan-line numbers are inclusive. If programmed to the same values, that single line defines the region in question.</p> <p>Always place an even number of MI_LOAD_SCAN_LINES_EXCL/INCL at a time into the ring buffer. If only a single MI_LOAD_SCAN_LINES_EXCL/INCL is desired, just add a second identical MI_LOAD_SCAN_LINES_EXCL/INCL command.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	13h MI_LOAD_SCAN_LINES_EXCL
		Format:	OpCode
	22	Reserved	
		Format:	MBZ
	21:19	Display Pipe Select	
		Format:	U3
This field selects which Display Engine (pipe) this command is targeting.			
Value		Name	
0h		Display Pipe A	
1h		Display Pipe B	
2h		Reserved	
3h		Reserved	
4h	Display Pipe C		
5h	Reserved		
18:17	Reserved		
	16:6	Reserved	
Format:		MBZ	



MI_LOAD_SCAN_LINES_EXCL				
	5:0	DWord Length		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2. Excludes DWord (0,1).</td> </tr> </table>	Default Value:	0h
Default Value:	0h			
Format:	=n Total Length - 2. Excludes DWord (0,1).			
1	31:29	Reserved		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	28:16	Start Scan Line Number		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>U13 In scan lines, where scan line 0 is the first line of the display frame.</td> </tr> </table>	Format:	U13 In scan lines, where scan line 0 is the first line of the display frame.
		Format:	U13 In scan lines, where scan line 0 is the first line of the display frame.	
		<table border="1" style="width: 100%;"> <tr> <td>Range: [0, Display Buffer height in lines-1]</td> </tr> </table>	Range: [0, Display Buffer height in lines-1]	
	Range: [0, Display Buffer height in lines-1]			
	This field specifies the starting scan line number of the Scan Line Window.			
	15:13	Reserved		
<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ	
Format:	MBZ			
12:0	End Scan Line Number			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>U13 In scan lines, where scan line 0 is the first line of the display frame.</td> </tr> </table>	Format:	U13 In scan lines, where scan line 0 is the first line of the display frame.	
	Format:	U13 In scan lines, where scan line 0 is the first line of the display frame.		
	This field specifies the ending scan line number of the Scan Line Window.			
<table border="1" style="width: 100%;"> <tr> <td>Range: [0, Display Buffer height in lines-1]</td> </tr> </table>	Range: [0, Display Buffer height in lines-1]			
Range: [0, Display Buffer height in lines-1]				



MI_LOAD_SCAN_LINES_INCL

MI_LOAD_SCAN_LINES_INCL			
Source:	BlitterCS		
Length Bias:	2		
<p>The MI_LOAD_SCAN_LINES_INCL command is used to initialize the Scan Line Window registers for a specific Display Engine. If the display refresh is <i>within</i> this window the Display Engine asserts a signal that is used by the command parser to process the WAIT_FOR_EVENT command (i.e., the parser will wait while inside of the window). This command overrides the Scan Line Window defined by any previous MI_LOAD_SCAN_LINES_INCL or MI_LOAD_SCAN_LINES_EXCL commands targeting the specific display.</p> <p>Always place an even number of MI_LOAD_SCAN_LINES_EXCL/INCL at a time into the ring buffer. If only a single MI_LOAD_SCAN_LINES_EXCL/INCL is desired, just add a second identical</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	12h MI_LOAD_SCAN_LINES_INCL
		Format:	OpCode
	22	Reserved	
	Format:		MBZ
	21:19	Display Pipe Select	
		Format: U3	
This field selects which Display Engine (pipe) this command is targeting.			
Value		Name	
0h		Display Pipe A	
1h		Display Pipe B	
2h, 3h		Reserved	
4h		Display Pipe C	
5h	Reserved		
6h, 7h	Reserved		
18:17	Reserved		
16:6	Reserved		
	Format: MBZ		
5:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n Total Length - 2	



MI_LOAD_SCAN_LINES_INCL		
1	31:16	Start Scan Line Number Format: U16 In scan lines, where scan line 0 is the first line of the display frame. This field specifies the starting scan line number of the Scan Line Window. Range: [0, Display Buffer height in lines-1]
	15:0	End Scan Line Number Format: U16 In scan lines, where scan line 0 is the first line of the display frame. This field specifies the ending scan line number of the Scan Line Window. Range: [0, Display Buffer height in lines-1]



MI_LOAD_SCAN_LINES_INCL

MI_LOAD_SCAN_LINES_INCL			
Source:	RenderCS		
Length Bias:	2		
<p>The MI_LOAD_SCAN_LINES_INCL command is used to initialize the Scan Line Window registers for a specific Display Engine. If the display refresh is within this window the Display Engine asserts a signal that is used by the command parser to process the WAIT_FOR_EVENT command (i.e., the parser will wait while inside the window). This command overrides the Scan Line Window defined by any previous MI_LOAD_SCAN_LINES_INCL or MI_LOAD_SCAN_LINES_EXCL commands targeting the specific display.</p> <p>Always place an even number of MI_LOAD_SCAN_LINES_EXCL/INCL at a time into the ring buffer. If only a single MI_LOAD_SCAN_LINES_EXCL/INCL is desired, just add a second identical MI_LOAD_SCAN_LINES_EXCL/INCL command.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	12h MI_LOAD_SCAN_LINES_INCL
		Format:	OpCode
	22	Reserved	
		Format:	MBZ
	21:19	Display Pipe Select	
		Format:	U3
		This field selects which Display Engine (pipe) this command is targeting.	
		Value	Name
		0h	Display Pipe A
		1h	Display Pipe B
2h		Reserved	
3h		Reserved	
4h	Display Pipe C		
5h	Reserved		
18:17	Reserved		
16:6	Reserved		
	Format:	MBZ	
5:0	DWord Length		
	Default Value:	0h	
	Format:	=n Total Length - 2. Excludes DWord (0,1).	



MI_LOAD_SCAN_LINES_INCL		
1	31	Reserved
		Format: MBZ
	30	Reserved
		Default Value: 1h
		Format: Must Be One
	29	Reserved
		Format: MBZ
	28:16	Start Scan Line Number
		Format: U13 In scan lines, where scan line 0 is the first line of the display frame.
		Range: [0, Display Buffer height in lines-1]
This field specifies the starting scan line number of the Scan Line window.		
15:13	Reserved	
	Format: MBZ	
12:0	End Scan Line Number	
	Format: U13 In scan lines, where scan line 0 is the first line of the display frame.	
	Range: [0, Display Buffer height in lines-1]	
	This field specifies the ending scan line number of the Scan Line Window.	



MI_LOAD_URB_MEM

MI_LOAD_URB_MEM			
Source:	RenderCS		
Length Bias:	2		
The MI_LOAD_URB_MEM command requests from a memory location and stores that DWord to the URB.			
Programming Notes			
The command temporarily halts commands that will cause cycles down the 3D pipeline.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	2Ch MI_LOAD_URB_MEM
22:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Format:	=n	
	Total Length - 2. Excludes DWord (0,1).		
	Value	Name	
	2h	[Default]	
1	31:15	Reserved	
		Format:	MBZ
	14:2	URB Address This field specifies Bits 14:2 of the URB offset the DWord will be written in the URB. This command only supports writing below 32KB of the URB space.	
1:0	Reserved		
	Format:	MBZ	
2..3	63:6	Memory Address	
		Format:	GraphicsAddress[63:6]
	This field specifies the address of the location of where the value will be read from memory. The value must be in the first DW location of the cache line. Range = GraphicsVirtualAddress[47:6] GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].		
5:0	Reserved		
	Format:	MBZ	



MI_MATH

MI_MATH		
Source:	BlitterCS	
Length Bias:	2	
<p>The MI_MATH command allows software to send instructions to the ALU in the Command Streamer. This command is the means by which the ALU is accessed. ALU instructions form the data payload of the MI_MATH command. An ALU instruction takes one DWord in size. The MI_MATH DWord Length is programmed based on the number of ALU instructions included, limited only by the max DWord Length supported. When the command streamer parses an MI_MATH command, it sends the included ALU instructions to the ALU. The ALU processes any instruction in a single clock. See the ALU section for more details.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 0h MI_COMMAND
		Format: OpCode
	28:23	MI Command Opcode
Default Value: 1Ah MI_MATH		
Format: OpCode		
22:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Default Value: 0h	
	Format: =n Total Length - 2. Excludes DWord (0,1).	
1	31:0	ALU INSTRUCTION 1
		Format: Table Entry
2	31:0	ALU INSTRUCTION 2
		Format: Table Entry
3..n	31:0	ALU INSTRUCTION n
		Format: Table Entry



MI_MATH

MI_MATH		
Source:	VideoCS	
Length Bias:	2	
<p>The MI_MATH command allows software to send instructions to the ALU in the Command Streamer. This command is the means by which the ALU is accessed. ALU instructions form the data payload of the MI_MATH command. An ALU instruction takes one DWord in size. The MI_MATH DWord Length is programmed based on the number of ALU instructions included, limited only by the max DWord Length supported. When the command streamer parses an MI_MATH command, it sends the included ALU instructions to the ALU. The ALU processes any instruction in a single clock. See the ALU section for more details.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 0h MI_COMMAND
		Format: OpCode
	28:23	MI Command Opcode
Default Value: 1Ah MI_MATH		
Format: OpCode		
22:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Default Value: 0h	
	Format: =n Total Length - 2. Excludes DWord (0,1).	
1	31:0	ALU INSTRUCTION 1
		Format: Table Entry
2	31:0	ALU INSTRUCTION 2
		Format: Table Entry
3..n	31:0	ALU INSTRUCTION n
		Format: Table Entry



MI_MATH

MI_MATH		
Source:	VideoEnhancementCS	
Length Bias:	2	
<p>The MI_MATH command allows software to send instructions to the ALU in the Command Streamer. This command is the means by which the ALU is accessed. ALU instructions form the data payload of the MI_MATH command. An ALU instruction takes one DWord in size. The MI_MATH DWord Length is programmed based on the number of ALU instructions included, limited only by the max DWord Length supported. When the command streamer parses an MI_MATH command, it sends the included ALU instructions to the ALU. The ALU processes any instruction in a single clock. See the ALU section for more details.</p>		
DWord	Bit	Description
0	31:29	Command Type Default Value: 0h MI_COMMAND Format: OpCode
		MI Command Opcode Default Value: 1Ah MI_MATH Format: OpCode
		Reserved Format: MBZ
	28:23	DWord Length Default Value: 0h Format: =n Total Length - 2. Excludes DWord (0,1).
22:8		
7:0		
1	31:0	ALU INSTRUCTION 1 Format: Table Entry
2	31:0	ALU INSTRUCTION 2 Format: Table Entry
3..n	31:0	ALU INSTRUCTION n Format: Table Entry



MI_MATH

MI_MATH		
Source:	RenderCS	
Length Bias:	2	
<p>The MI_MATH command allows SW to send instruction to ALU in Render Command Streamer. MI_MATH command is the means by which ALU can be accessed. ALU instructions form the data payload of MI_MATH command, ALU instruction is dword in size. MI_MATH Dword Length should be programmed based on the number of ALU instruction packed, max number is limited by the max Dword Length supported. When MI_MATH command is parsed by command streamer it outputs the payload dwords (ALU instructions) to the ALU. ALU takes single clock to process any given instruction. Refer to B-spec "Command Streamer (CS) ALU Programming" section in Command Streamer Programming.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 0h MI_COMMAND Format: OpCode
	28:23	MI Command Opcode
		Default Value: 1Ah MI_MATH Format: OpCode
22:8	Reserved	
		Format: MBZ
7:0	DWord Length	
	Default Value:	0h
	Format:	=n Total Length - 2. Excludes DWord (0,1).
1	31:0	ALU INSTRUCTION 1
		Format: Table Entry
2	31:0	ALU INSTRUCTION 2
		Format: Table Entry
3..n	31:0	ALU INSTRUCTION n
		Format: Table Entry



MI_NOOP

MI_NOOP			
Source:	VideoEnhancementCS		
Length Bias:	1		
<p>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).</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	00h MI_NOOP
		Format:	OpCode
	22	Identification Number Register Write Enable	
		Format:	Enable
		<p>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 operation" function.</p>	
		Value	Name
1			Write th NOP_ID Register
0		Do not write the NOP_ID register	
21:0	Identification Number		
	Format:	U22	
<p>This field contains a 22-bit number which can be written to the MI NOPID register.</p>			



MI_NOOP

MI_NOOP									
Source:	BlitterCS								
Length Bias:	1								
<p>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).</p>									
DWord	Bit	Description							
0	31:29	Command Type							
		Default Value: 0h MI_COMMAND							
	28:23	MI Command Opcode							
		Default Value: 0h MI_NOOP							
22		Identification Number Register Write 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 operation" function.							
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Do not write the NOP_ID register.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Write the NOP_ID register.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	Do not write the NOP_ID register.	1h
Value	Name	Description							
0h	Disable	Do not write the NOP_ID register.							
1h	Enable	Write the NOP_ID register.							
21:0		Identification Number							
		Format: U22							
This field contains a 22-bit number which can be written to the MI NOPID register.									



MI_NOOP

MI_NOOP											
Source:	RenderCS										
Length Bias:	1										
<p>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).</p>											
Performance											
<p>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.</p>											
DWord	Bit	Description									
0	31:29	Command Type Default Value: 0h MI_COMMAND									
	28:23	MI Command Opcode Default Value: 0h MI_NOOP									
	22	Identification Number Register Write 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 operation" function. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Do not write the NOP_ID register.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Write the NOP_ID register.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	Do not write the NOP_ID register.	1h	Enable	Write the NOP_ID register.
	Value	Name	Description								
0h	Disable	Do not write the NOP_ID register.									
1h	Enable	Write the NOP_ID register.									
21:0	Identification Number Format: U22 This field contains a 22-bit number which can be written to the MI NOPID register.										



MI_NOOP

MI_NOOP			
Source:	VideoCS		
Length Bias:	1		
<p>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).</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value: 0h MI_COMMAND	
		Format: OpCode	
	28:23	MI Command Opcode	
		Default Value: 00h MI_NOOP	
		Format: OpCode	
	22	Identification Number Register Write 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 operation" function.	
		Value	Name
		1	Write the NOP_ID register.
	21:0	Identification Number	
Format: U22			
This field contains a 22-bit number which can be written to the MI NOPID register.			



MI_PREDICATE

MI_PREDICATE			
Source:		RenderCS	
Length Bias:		1	
DWord	Bit	Description	
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:	MBZ
	7:6	Load Operation	
		This field controls if/how the Predicate state bit is modified.	
		Value	Name Description
		0h	KEEP
1h		Reserved	
2h		LOAD	The Predicate state bit is loaded with the combine operation result.
3h	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	SET	The combine operation output the compare result unmodified.
	1h	AND	The combine operation outputs the AND of the compare result and the current Predicate state bit.
	2h	OR	The combine operation outputs the OR of the compare result and the current Predicate state bit.
3h	XOR	The combine operation outputs the XOR of the compare result and the current Predicate state bit.	
2	Reserved		
	Format:	MBZ	



MI_PREDICATE

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.	
	Value	Name
	Description	
	0h	TRUE
1h	FALSE	
2h	SRCS_EQUAL	
3h	DELTAS_EQUAL	



MI_REPORT_HEAD

MI_REPORT_HEAD		
Source:	VideoEnhancementCS	
Length Bias:	1	
<p>The MI_REPORT_HEAD command causes the Head Pointer value of the ring buffer to be written to a cacheable (snooped) system memory location.</p> <p>When the Per-Process Virtual Address Space and Execlist Enable bit is reset: The location written is relative to the address programmed in the Hardware Status Page Address Register. When the Execlist Enable is set, the head pointer will be reported to the PP HW Status Page.</p>		
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

MI_REPORT_HEAD		
Source:	BlitterCS	
Length Bias:	1	
<p>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.</p> <p>When the Execlist Enable bit is reset: The location written is relative to the address programmed in the Hardware Status Page Address Register.</p>		
Programming Notes		
<p>This command must not be executed from a Batch Buffer (Refer to the description of the HWS_PGA register).</p> <p>When the Execlist Disable is clear, the head pointer will be reported to the PP HW Status Page.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 0h MI_COMMAND
	28:23	MI Command Opcode
		Default Value: 07h MI_REPORT_HEAD
	22:0	Reserved
		Format: MBZ



MI_REPORT_HEAD

MI_REPORT_HEAD		
Source:	RenderCS	
Length Bias:	1	
<p>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. When Execlist Enable is set, the head pointer will be reported to the PP HW Status Page. The location written is relative to the address programmed in the Hardware Status Page Address Register.</p>		
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

MI_REPORT_HEAD		
Source:	VideoCS	
Length Bias:	1	
<p>The MI_REPORT_HEAD command causes the Head Pointer value of the ring buffer to be written to a cacheable (snooped) system memory location. When the Per-Process Virtual Address Space and Execlist Enable bits are reset, the location written is relative to the address programmed in the Hardware Status Page Address Register. When the Execlist Enable is set, the head pointer will be reported to the PP HW Status Page.</p>		
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_PERF_COUNT

MI_REPORT_PERF_COUNT

Source: RenderCS
 Length Bias: 2

The MI_REPORT_PERF_COUNT command causes the GFX hardware to write out a snap-shot of performance counters to the address specified in this command along with constant ID field supplied and the time-stamp counter. This write is required to be treated as a cacheable write irrespective of GTT entry memory type. This command is specific to the render engine.

Programming Notes

GTT_SELECT must not be set to 1 (i.e. GGTT) when MI_REPORT_PERF_COUNT command is programmed in a non-privileged batch buffer. Refer to the "User Mode Privileged commands" Table in MI_BATCH_BUFFER_START command section for more details.
 All batch buffers in PPGTT are considered as Non-privileged.

DWord	Bit	Description	
0	31:29	Command Type	
		Default Value: 0h MI_COMMAND	
		Format: OpCode	
	28:23	MI Command Opcode	
		Default Value: 28h MI_REPORT_PERF_COUNT	
		Format: OpCode	
	22:6	Reserved	
		Format: MBZ	
5:0	DWord Length		
	Format: =n		
	Total Length - 2		
	Value	Name	
	2h	Excludes DWord (0,1) [Default]	
1..2	63:6	Memory Address	
		Format: GraphicsAddress[63:6]	
		This field specifies 64B aligned GFX MEM address where the chap counter values are reported. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47]	
		Programming Notes	
	This field is ignored if "Report to OABUFFER" bit is set.		
5	5	Reserved	
		Format: MBZ	



MI_REPORT_PERF_COUNT					
	4	<p>Core Mode Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U1</td> </tr> </table> <p>This bit is set then the address will be offset by the Core ID: If Core ID 0, then there is no offset. If Core ID 1, then the Memory is offset by the size of the data(64b).</p>	Format:	U1	
	Format:	U1			
	3:1	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ				
0	<p>Use Global GTT</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Boolean</td> </tr> </table> <p>This field when set (i.e. bit = 1) selects the GGTT for address translation. When this bit is 0 (default value), HW should use PGTT for address translation.</p>	Format:	Boolean		
Format:	Boolean				
3	31:0	<p>Report ID</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U32</td> </tr> </table> <p>This field specifies the ID provided by SW for a given report command. It can be tracked to use different flavors of these reports based on where in command-stream they are inserted. This field is reported only when Counter Select Field is 0.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>If a privilege access violation occurs, the REPORT ID field in the report generated by the next legitimate MI_REPORT_PERF_COUNT will be corrupted.</p>	Format:	U32	Programming Notes
Format:	U32				
Programming Notes					



MI_RS_CONTEXT

MI_RS_CONTEXT			
Source:	RenderCS		
Length Bias:	1		
The MI_RS_CONTEXT command is used to force a resource streamer context save or restore.			
Programming Notes			
This command must not be used/programmed in Execution List mode of scheduling.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	0Fh MI_RS_CONTEXT
		Format:	OpCode
	22:1	Reserved	
		Format:	MBZ
	0	Resource Streamer Save	
		Format:	U1
This bit specifies whether the MI_RS_CONTEXT command will cause the resource streamer context to be saved or restored.			
Value		Name	Description
0h		Restore	Resource Streamer context is restored
1h	Save	Resource Streamer context is saved	



MI_RS_CONTROL

MI_RS_CONTROL						
Source:	RenderCS					
Length Bias:	1					
The MI_RS_CONTROL command is used to start or stop the Resource Streamer.						
Programming Notes						
<ul style="list-style-type: none"> • This command must be programmed only inside a Resource Streamer enabled batch buffer. • This command provides means to selectively disable or enable Resource Streamer for set of commands in a Resource Streamer enabled batch buffer • On re-enabling the Resource Streamer through this command, command streamer will start Resource Streamer on the next non-sync command of the batch buffer. • This command status is render context save/restored during context switching. • The scope of MI_RS_CONTROL is within the batch buffer it is programmed, it doesn't get carried to the following chained batch buffer or second level batch buffer. RS control status goes back to default mode of Resource Streamer Enabled on all batch buffer arbitration boundaries. Batch buffer arbitration boundaries includes calling a chained or a second level batch buffer through MI_BATCH_BUFFER_START command or terminating a batch buffer through MI_BATCH_BUFFER_END command. • Example: <ol style="list-style-type: none"> 1. MI_BATCH_START (Primary batch buffer with RS enable) 2. Command 1 --> CS starts RS 3. Command 2 4. : 5. MI_RS_CONTROL (stop option) -> RS will stop on this command, CS sets RS control status to STOP. 6. Command 3 7. MI_BATCH_START (2nd level batch with RS enable not set, RS control status gets reset to default status of START) 8. : 9. MI_BATCH_END (Second Level Batch End) 10. Command 4 --> CS starts RS here as RS control flag gets reset to START at step-7 11. MI_BATCH_BUFFER_END 						
DWord	Bit	Description				
0	31:29	Command Type <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>0h MI_COMMAND</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	0h MI_COMMAND	Format:	OpCode
Default Value:	0h MI_COMMAND					
Format:	OpCode					



MI_RS_CONTROL											
	28:23	MI Command Opcode									
		Default Value: 06h MI_RS_CONTROL									
		Format: OpCode									
	22:1	Reserved									
		Format: MBZ									
	0	Resource Streamer Control									
		Format: U1									
		This bit specifies whether the command is starting or stopping the Resource Streamer.									
		<table border="1"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>0h</td><td>Stop</td><td>Stop and disable the Resource Streamer</td></tr><tr><td>1h</td><td>Start</td><td>Start and enable the Resource Streamer</td></tr></tbody></table>	Value	Name	Description	0h	Stop	Stop and disable the Resource Streamer	1h	Start	Start and enable the Resource Streamer
		Value	Name	Description							
0h	Stop	Stop and disable the Resource Streamer									
1h	Start	Start and enable the Resource Streamer									



MI_RS_STORE_DATA_IMM

MI_RS_STORE_DATA_IMM			
Source:	RenderCS		
Length Bias:	2		
The MI_RS_STORE_DATA_IMM command requests a write of the DWord constant supplied in the packet to the specified Memory Address.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	2Bh
		Format:	OpCode
			MI_RS_STORE_DATA_IMM
22	Reserved		
Format:	MBZ		
21	Reserved		
20:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	2h	
	Format:	=n Total Length - 2. Excludes DWord (0,1).	
1..2	63:2	Destination Address	
		Format:	GraphicsAddress[63:2]
		This field specifies Bits 47:2 of the Address where the DWord will be stored. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47]. When render engine is PPGTT enabled this Address is translated using PPGTT, else GGTT is used for translation.	
	1	Reserved	
Format:	MBZ		
0	Core Mode Enable		
		If this bit is set then the address will be offset by the Core ID: If Core ID 0, then there is no offset If Core ID 1, then the Memory is offset by the size of the data.	



MI_RS_STORE_DATA_IMM		
3	31:0	Data DWord 0
		<table border="1"><tr><td>Format:</td><td>U32</td></tr></table> <p>This field specifies the DWord value to be written to the targeted location.</p>
Format:	U32	



MI_SEMAPHORE_SIGNAL

MI_SEMAPHORE_SIGNAL			
Source:	CommandStreamer		
Length Bias:	2		
Description			
<p>This command is used to signal the target engine stating the memory semaphore update occurrence to one of its contexts with Target Context ID. MI_SEMPHORE_SIGNAL and MI_SEMAPHORE_WAIT together replace the MI_SEMAPHORE_MBOX command. MI_ATOMIC (non-posted) command will be programmed prior to this command to update the semaphore data in memory.</p>			
Workaround			
Workaround: Post-Sync operation bit must not be set when Target Engine Select is set to RCS.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	1Bh MI_SEMAPHORE_SIGNAL
		Format:	OpCode
	22	Reserved	
		Format:	MBZ
	21	Post-Sync Operation	
		Source:	RenderCS
Value		Name	Description
0h		No Post Sync Operation	Command is executed as usual.
1h		Post Sync Operation	MI_SEMAPHORE_SIGNAL command is executed as a pipelined PIPE_CONTROL flush command with Semaphore Signal as post sync operation. Flush completion only guarantees the workload prior to this command is pushed till Windower unit and completion of any outstanding flushes issued prior to this command.
Programming Notes			
<p>Any desired pipeline flush operation can be achieved by programming PIPE_CONTROL command prior to this command.</p> <p>When this bit is set Command Streamer sends a flush down the pipe and the atomic operation is saved as post sync operation. Command streamer goes on executing the following</p>			



MI_SEMAPHORE_SIGNAL															
	<p>commands. Atomic operation saved as post sync operation is executed at some point later on completion of corresponding flush issued.</p> <p>When this bit is set atomic semaphore signal operation will be out of order with rest of the MI commands programmed in the ring buffer or batch buffer, it will be in order with respect to the post sync operations resulting due to PIPE_CONTROL command.</p>														
21	<p>Reserved</p> <table border="1"> <tr> <td>Source:</td> <td>BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS	Format:	MBZ										
Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS														
Format:	MBZ														
20:19	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ														
18	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ														
17:15	<p>Target Engine Select This field selects the target engine to which SIGNAL will be send to.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>RCS</td> </tr> <tr> <td>1h</td> <td>VCS0</td> </tr> <tr> <td>2h</td> <td>BCS</td> </tr> <tr> <td>3h</td> <td>VECS</td> </tr> <tr> <td>4h</td> <td>VCS1</td> </tr> <tr> <td>5h,6h,7h</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	0h	RCS	1h	VCS0	2h	BCS	3h	VECS	4h	VCS1	5h,6h,7h	Reserved
Value	Name														
0h	RCS														
1h	VCS0														
2h	BCS														
3h	VECS														
4h	VCS1														
5h,6h,7h	Reserved														
14:8	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ														
7:0	<p>DWord Length</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h Excludes DWord (0,1)</td> </tr> <tr> <td>Format:</td> <td>=n</td> </tr> </table> <p>Total Length - 2</p>	Default Value:	0h Excludes DWord (0,1)	Format:	=n										
Default Value:	0h Excludes DWord (0,1)														
Format:	=n														



MI_SEMAPHORE_SIGNAL					
1	31:0	<p>Target Context ID</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center; color: #0070C0;">Description</th> </tr> </thead> <tbody> <tr> <td> <p>In execlist based scheduling this field contains the Context ID corresponding to the context of the target engine that this command is signaling. Target engine waiting on MI_SEMAPHORE_WAIT in signal mode will re-fetch the data from memory or comparison if its context ID is same as this signaled Context ID. When execlists are enabled, Target engine on receiving this Context ID sends message to the SHIM if it doesn't have the context with the same Context ID running. Message send to SHIM carries the Context ID which will be looked at by UC for rescheduling the signaled Context ID. Target engine waiting on MI_SEMAPHORE_WAIT in signal mode will fetch data from memory for comparison on receiving signal irrespective of the context id received.</p> </td> </tr> <tr> <td> <p>In ring buffer mode of scheduling this field doesn't have any relevance.</p> </td> </tr> </tbody> </table>	Description	<p>In execlist based scheduling this field contains the Context ID corresponding to the context of the target engine that this command is signaling. Target engine waiting on MI_SEMAPHORE_WAIT in signal mode will re-fetch the data from memory or comparison if its context ID is same as this signaled Context ID. When execlists are enabled, Target engine on receiving this Context ID sends message to the SHIM if it doesn't have the context with the same Context ID running. Message send to SHIM carries the Context ID which will be looked at by UC for rescheduling the signaled Context ID. Target engine waiting on MI_SEMAPHORE_WAIT in signal mode will fetch data from memory for comparison on receiving signal irrespective of the context id received.</p>	<p>In ring buffer mode of scheduling this field doesn't have any relevance.</p>
Description					
<p>In execlist based scheduling this field contains the Context ID corresponding to the context of the target engine that this command is signaling. Target engine waiting on MI_SEMAPHORE_WAIT in signal mode will re-fetch the data from memory or comparison if its context ID is same as this signaled Context ID. When execlists are enabled, Target engine on receiving this Context ID sends message to the SHIM if it doesn't have the context with the same Context ID running. Message send to SHIM carries the Context ID which will be looked at by UC for rescheduling the signaled Context ID. Target engine waiting on MI_SEMAPHORE_WAIT in signal mode will fetch data from memory for comparison on receiving signal irrespective of the context id received.</p>					
<p>In ring buffer mode of scheduling this field doesn't have any relevance.</p>					



MI_SEMAPHORE_WAIT

MI_SEMAPHORE_WAIT		
Source:	CommandStreamer	
Length Bias:	2	
Description		
<p>This command supports memory based Semaphore WAIT. Memory based semaphores will be used for synchronization between the Producer and the Consumer contexts. Producer and Consumer Contexts could be running on different engines or on the same engine inside GT. Running on the same engine is only possible when execlists are enabled. Producer Context implements a Signal and Consumer context implements a Wait. Command Streamer on parsing this command fetches data from the Semaphore Address mentioned in this command and compares it with the inline Semaphore Data Dword.</p> <ul style="list-style-type: none"> • If comparison passes, the command streamer moves to the next command. • When execlists are enabled, if comparison fails Command streamer switches out the context. Context switch can be inhibited by setting "Inhibit Synchronous Context Switch" in CTXT_SR_CTL register. • In ring buffer mode of scheduling or Execlist with "Inhibit Synchronous context Switch", if comparison fails, Command Streamer evaluates the Compare Operation based on the Wait Mode until the compare operation is true or Wait is canceled by SW. • Exec-List Scheduling: CS generates semaphore wait interrupt to the scheduler when MI_SEMAPHORE_WAIT command is un-successful and when "Inhibit Synchronous Context Switch" is set. Scheduler can use this interrupt to preempt the context waiting on semaphore wait. <p>MI_SEMPHORE_SIGNAL and MI_SEMAPHORE_WAIT together replace the MI_SEMAPHORE_MBOX command.</p>		
Workaround		
<p>Workaround: [All Command Streamers][Ring Buffer Mode of Scheduling]: MI_SEMAPHORE_WAIT command must be always programmed with "Wait Mode" set to "Polling Mode" Or MI_SEMAPHORE_WAIT command with "Wait Mode" set to "Polling Mode" can be programmed when "Semaphore Wait Event IDLE message Disable" bit in "RC_PSMI_CTRL" register is set to disable Idle messaging on unsuccessful MI_SEMPAHORE_WAIT.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value:
	Format:	OpCode
	28:23	MI Command Opcode
		Default Value:
	Format:	OpCode
22	Memory Type	This bit will be ignored and treated as if clear when executing from a non-privileged batch buffer. It is allowed for this bit to be clear when executing this command from a privileged (secure) batch buffer. This bit <i>must</i> be 1 if the Per Process GTT Enable bit is clear.



MI_SEMAPHORE_WAIT			
	Value	Name	Description
	0h	Per Process Graphics Address	
	1h	Global Graphics Address	This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.
21:18	Reserved		
	Format:		MBZ
17	Reserved		
	Format:		MBZ
16	Reserved		
15	Wait Mode		
	This bit specifies the WAIT behavior when the semaphore comparison fails and before the context is switched out.		
	Value	Name	Description
	1h	Polling Mode	In this mode HW periodically reads the semaphore data from memory for comparison until it is context switched out. Periodicity will be mentioned in a SEMA_WAIT_POLL register.
	0h	Signal Mode	In this mode HW will reacquire the semaphore data from memory on receiving SIGNAL with the same Context ID. In ring buffer mode of scheduling Context ID associated with SIGNAL is ignored and always treated as a match.
14:12	Compare Operation		
	This field specifies the operation that will be executed to create the result that will either allow the context to continue or wait.		
	SAD = Semaphore Address Data SDD = Semaphore Data Dword		
	Value	Name	Description
	0h	SAD_GREATER_THAN_SDD	If Indirect fetched data is greater than inline data then continue.
	1h	SAD_GREATER_THAN_OR_EQUAL_SDD	If Indirect fetched data is greater than or equal to inline data then continue.
	2h	SAD_LESS_THAN_SDD	If Indirect fetched data is less than inline data then continue.
	3h	SAD_LESS_THAN_OR_EQUAL_SDD	If Indirect fetched data is less than or equal to inline data then continue.
	4h	SAD_EQUAL_SDD	If Indirect fetched data is equal to inline data then continue.



MI_SEMAPHORE_WAIT			
	5h	SAD_NOT_EQUAL_SDD	If Indirect fetched data is not equal to inline data then continue.
	6h	Reserved	
	7h	Reserved	
	11:8	Reserved	
	Format:		MBZ
	7:0	DWord Length	
		Format: =n Total Length - 2	
		Value	Name
	2h	Excludes DWord (0,1) [Default]	
1	31:0	Semaphore Data Dword	
		Format:	U32
This Data dword is supplied by software to control execution of the command buffer. This value is used as part of the comparison to result in waiting or continuing in the command parser if enabled.			
2..3	63:2	Semaphore Address	
		Format:	GraphicsAddress[63:2]
This field is the Graphics Memory Address of the 32-bit value for the semaphore. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form.			
	1:0	Reserved	
		Format:	MBZ



MI_SET_CONTEXT

MI_SET_CONTEXT			
Source:	RenderCS		
Length Bias:	2		
<p>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. 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). All context is saved and restored from a GGTT space. This command does not initiate any interrupt due to context switch of any kind and does not support any workaround batch buffer or indirect context offset feature.</p>			
Programming Notes			
This command is legal only if Execlist Enable in the GFX_MODE register is reset. Otherwise, execlists must be used to switch context in lieu of MI_SET_CONTEXT.			
For ring buffer mode, the first 128B(2 cache lines) of the context image are saved as zeros.			
This command needs to be always followed by a single MI_NOOP instruction to workaround a silicon issue.			
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.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	18h MI_SET_CONTEXT
		Format:	OpCode
22:8	Reserved		
	Format:	MBZ	



MI_SET_CONTEXT					
	7:0	DWord Length			
	<table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2. Excludes DWord (0,1).</td> </tr> </table>		Default Value:	0h	Format:
Default Value:	0h				
Format:	=n Total Length - 2. Excludes DWord (0,1).				
1	31:12	Logical Context Address			
		Format: GraphicsAddress[31:12]LogicalContext			
	Description				
	<p>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 this new context, the device will save the existing context as required. After the context switch operation completes, this address will be loaded into the associated CCID register.</p>				
	<p>This field needs to be 4KB aligned virtual address.</p>				
	11:10	Reserved			
	Format: MBZ				
	9	Reserved			
	Format: MBZ				
	8	Reserved, Must be 1			
Format: Must Be One					
7:5	Reserved				
Format: MBZ					
4	Core Mode Enable				
Format: Enable					
<p>If set the Context Image will be offset based off the Core ID: If Core ID 0, no offset If Core ID 1, 36KB Offset</p>					
3	Resource Streamer State Save Enable				
Format: Enable					
<p>If set, the resource streamer 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 switching away from this context (as part of a subsequent MI_SET_CONTEXT command).</p>					



MI_SET_CONTEXT			
2	<p>Resource Streamer State Restore Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, the resource streamer state identified in the Logical Context Data section of the Memory Data Formats chapter is loaded (or restored) as part of switching to this logical context. This bit affects the switch (if required) to the context specified in Logical Context Address. This bit will also be stored in the associated CCID register to control a subsequent context save operation when switching to this context (as part of a subsequent ring buffer switch).</p>	Format:	Enable
Format:	Enable		
1	<p>Force Restore</p> <p>When switching to this logical context a comparison between Logical Context Address and the contents 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.</p>		
0	<p>Restore Inhibit</p> <p>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.</p>		



MI_SET_PREDICATE

MI_SET_PREDICATE			
Source:	RenderCS		
Length Bias:	1		
Description			
<p>This command sets the Predication Check for the subsequent commands in the command buffer except for MI_SET_PREDICATE itself. Render Command Streamer NOOPs the following commands based on the PREDICATE_ENABLE from MI_SET_PREDICATE, MI_SET_PREDICATE_RESULT and MI_SET_PREDICATE_RESULT_2 status. Resource Streamer doesn't take any action of parsing MI_SET_PREDICATE, this command is similar to any other command which is not meant for resource streamer.</p> <p>Executing MI_SET_PREDICATE command sets PREDICATE_ENABLE bits in MI_MODE register, MI_MODE register gets render context save restored.</p>			
Programming Notes			
<ul style="list-style-type: none"> MI_SET_PREDICATE predication scope must be confined within a Batch Buffer to set of commands. MI_SET_PREDICATE with Predicate Enable Must always have a corresponding MI_SET_PREDICATE with Predicate Disable within the same Batch Buffer. MI_ARB_CHK command must be programmed outside the Predication Scope of MI_SET_PREDICATE. MI_SET_PREDICATE Predication Scope must not involve any RC6 triggering events. 			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	01h MI_SET_PREDICATE
		Format:	OpCode
	22:4	Reserved	
		Format:	MBZ
	3:0	PREDICATE ENABLE	
		This field sets the predication logic in render command streamer when parsed. Predicate Disable is the default mode of operation.	
		Value	Description
		0h	NOOP Never Predication is Disabled and RCS will process commands as usual.
1h	NOOP on Result2 clear Following Commands will be NOOPED by RCS only if the MI_PREDICATE_RESULT_2 is clear.		



MI_SET_PREDICATE		
2h	NOOP on Result2 set	Following Commands will be NOOPED by RCS only if the MI_PREDICATE_RESULT_2 is set.
3h	NOOP on Result clear	Following Commands will be NOOPED by RCS only if the MI_PREDICATE_RESULT is clear.
4h	NOOP on Result set	Following Commands will be NOOPED by RCS only if the MI_PREDICATE_RESULT is set.
5h	Execute when one slice enabled.	Following Commands will be Executed by RCS only when one slice is enabled.
6h	Execute when two slices are enabled.	Following Commands will be Executed by RCS only when two slices are enabled.
7h	Execute when three slices are enabled.	Following Commands will be Executed by RCS only when all the three slices are enabled.
8h-Ah	Reserved	
Bh, Ch	Reserved	
Dh, Eh	Reserved	
Fh	NOOP Always	Following Commands will be NOOPED by RCS unconditionally.



MI_STORE_DATA_IMM

MI_STORE_DATA_IMM			
Source:	CommandStreamer		
Length Bias:	2		
<p>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).</p> <p>This command supports writing to multiple consecutive dwords or qwords memory locations from the starting address.</p>			
Programming Notes			
<ul style="list-style-type: none"> This command should not be used within a "non-privilege" batch buffer to access global virtual space, doing so will be treated as privilege access violation. Refer "User Mode Privilege Command" in MI_BATCH_BUFFER_START command section to know HW behavior on encountering privilege access violation. This command can be used within ring buffers and/or privilege batch buffers to access global virtual space. 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 Opcode	
		Default Value:	20h MI_STORE_DATA_IMM
		Format:	OpCode
22	Use Global GTT		
	Format:	Boolean	
<p>If set, this command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer. If clear, the PPGTT will be used. It is allowed for this bit to be clear when executing this command from a privileged (secure) batch buffer. This bit must be '1' if the Per Process GTT Enable bit is clear.</p>			



MI_STORE_DATA_IMM														
	<p>21 Store Qword</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Boolean</td> </tr> </table> <p>If set, this command generates Qword writes to memory, two "Data Dword" are paired to form a Qword. Number of qwords generated depends upon the number of "Data Dword" programmed in the command. If 'x' number of "Data Dwords" are programmed in this command it results in "x/2" qword writes to memory.</p> <p>If reset this command generates Dwords writes to memory. Number of dwords generated depends upon the number of "Data Dword" programmed in the command. If 'x' number of "Data Dwords" are programmed in this command it results in "x" dword writes to memory.</p>	Format:	Boolean											
	Format:	Boolean												
	<p>20:10 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ											
	Format:	MBZ												
<p>9:0 DWord Length</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>=n Total Length - 2. Excludes DWord (0,1)</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>2h</td> <td>Store Dword [Default]</td> </tr> <tr> <td>3h</td> <td>Store Qword</td> </tr> </tbody> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 80%;">Programming Notes</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td>DWord Length programmed must not exceed 0x3FE.</td> <td></td> </tr> <tr> <td>If RS is enabled in the batch buffer, then the value of this field must not exceed 0x3F.</td> <td>RenderCS</td> </tr> </tbody> </table>	Format:	=n Total Length - 2. Excludes DWord (0,1)	Value	Name	2h	Store Dword [Default]	3h	Store Qword	Programming Notes	Source	DWord Length programmed must not exceed 0x3FE.		If RS is enabled in the batch buffer, then the value of this field must not exceed 0x3F.	RenderCS
Format:	=n Total Length - 2. Excludes DWord (0,1)													
Value	Name													
2h	Store Dword [Default]													
3h	Store Qword													
Programming Notes	Source													
DWord Length programmed must not exceed 0x3FE.														
If RS is enabled in the batch buffer, then the value of this field must not exceed 0x3F.	RenderCS													
<p>1..2 63:48 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ													
<p>47:2 Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:2]</td> </tr> </table> <p>GraphicsAddress is 64-bit value [63:0], but only a portion of it is used by hardware. The uppermost bits are ignored and MBZ. This field specifies Bits 47:2 of the Address where the DWord will be stored. As the store address must be DWord-aligned, Bits 1:0 of that address MBZ. This address must be 8B aligned for a store "QW" command.</p>	Format:	GraphicsAddress[47:2]												
Format:	GraphicsAddress[47:2]													
<p>1 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
Format:	MBZ													
<p>0 Core Mode Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U1</td> </tr> </table> <p>This bit is set then the address will be offset by the Core ID: If Core ID 0, then there is no offset If Core ID 1, then the Memory is offset by the size of the data(32b or 64b based off number of DW length).</p>	Format:	U1												
Format:	U1													



MI_STORE_DATA_IMM		
3	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).
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).



MI_STORE_DATA_INDEX

MI_STORE_DATA_INDEX			
Source:	VideoEnhancementCS		
Length Bias:	2		
<p>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).</p>			
Programming Notes			
<ul style="list-style-type: none"> • 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 Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	21h MI_STORE_DATA_INDEX
	22	Reserved	
Format:		MBZ	
21	Use Per-Process Hardware Status Page If this bit is set, this command will index into the per-process hardware status page at offset 0K from the LRCA. If clear, the Global Hardware Status Page will be indexed. This bit must be '0' if the Execlist Enable bit is clear.		
20:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1) = 2 for QWord	
	Format:	=n Total Length - 2	
1	31:12	Reserved	
		Format:	MBZ



MI_STORE_DATA_INDEX						
	11:2	Offset				
		Format: U10 Zero-based DWord offset into the HW status page				
		Format: GraphicsAddress[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 of this status page are reserved for special-purpose data storage - targeting these reserved locations via this command is UNDEFINED. For a QWord write, the offset is valid down to bit 3 only.				
		<table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>[16, 1023]</td><td></td></tr></tbody></table>	Value	Name	[16, 1023]	
Value	Name					
[16, 1023]						
	1:0	Reserved				
		Format: MBZ				
2	31:0	Data DWord 0				
		Format: U32				
		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				
		This field specifies the upper DWord value to be written to the targeted QWord location (DW 1).				



MI_STORE_DATA_INDEX

MI_STORE_DATA_INDEX		
Source:	BlitterCS	
Length Bias:	2	
<p>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).</p>		
Programming Notes		
<p>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.</p>		
DWord	Bit	Description
0	31:29	Command Type Default Value: 0h MI_COMMAND
	28:23	MI Command Opcode Default Value: 21h MI_STORE_DATA_INDEX
	22	Reserved Format: MBZ
	21	Use Per-Process Hardware Status Page If this bit is set, this command will index into the per-process hardware status page at offset 0K from the LRCA. If clear, the Global Hardware Status Page will be indexed. This bit must be '0' if the Execlst Enable bit is clear.
	20:8	Reserved Format: MBZ
	7:0	DWord Length Default Value: 1h Excludes DWord (0,1) = 1 for DWord, 2 for QWord Format: =n Total Length - 2
1	31:12	Reserved Format: MBZ
	11:2	Offset Format: U10 zero-based DWord offset into the HW status page.
		Format: HardwareStatusPageOffset[11:2]U32 This field specifies the offset (into the hardware status page) to which the data will be written.



MI_STORE_DATA_INDEX						
		<p>Note that the first few DWords of this status page are reserved for special-purpose data storage - targeting these reserved locations via this command is UNDEFINED. This address must be 8B aligned for a store "QW" command.</p> <table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>[16, 1023]</td><td></td></tr></tbody></table>	Value	Name	[16, 1023]	
Value	Name					
[16, 1023]						
	1:0	<p>Reserved</p> <table border="1"><tr><td>Format:</td><td>MBZ</td></tr></table>	Format:	MBZ		
Format:	MBZ					
2	31:0	<p>Data DWord 0</p> <table border="1"><tr><td>Format:</td><td>U32</td></tr></table> <p>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).</p>	Format:	U32		
Format:	U32					
3	31:0	<p>Data DWord 1</p> <table border="1"><tr><td>Format:</td><td>U32</td></tr></table> <p>This field specifies the upper DWord value to be written to the targeted QWord location (DW 1).</p>	Format:	U32		
Format:	U32					



MI_STORE_DATA_INDEX

MI_STORE_DATA_INDEX			
Source:	RenderCS		
Length Bias:	2		
<p>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).</p>			
Programming Notes			
<ul style="list-style-type: none"> • 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 Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	21h MI_STORE_DATA_INDEX
		Format:	OpCode
	22	Reserved	
	21	Use Per-Process Hardware Status Page If this bit is set, this command will index into the per-process hardware status page at offset 0K from the LRCA. If clear, the Global Hardware Status Page will be indexed. This bit must be 0 if the Execlist Enable bit is clear.	
	20:8	Reserved	
		Format: MBZ	
	7:0	DWord Length	
		Default Value:	1h
		Format:	=n Total Length - 2. Excludes DWord (0,1) = 1 for DWord, 2 for QWord.
1	31:12	Reserved	
		Format: MBZ	



MI_STORE_DATA_INDEX										
	11:2	Offset <table border="1"><tr><td>Format:</td><td>U10 zero-based DWord offset into the HW status page.</td></tr><tr><td>Format:</td><td>HardwareStatusPageOffset[11:2]U32</td></tr></table> <p>This field specifies the offset (into the hardware status page) to which the data will be written. Note that the first few DWords of this status page are reserved for special-purpose data storage - targeting these reserved locations via this command is UNDEFINED. This address must be 8B aligned for a store QW command.</p> <table border="1"><thead><tr><th style="text-align: center;">Value</th><th style="text-align: center;">Name</th></tr></thead><tbody><tr><td>[16, 1023]</td><td></td></tr></tbody></table>	Format:	U10 zero-based DWord offset into the HW status page.	Format:	HardwareStatusPageOffset[11:2]U32	Value	Name	[16, 1023]	
	Format:	U10 zero-based DWord offset into the HW status page.								
Format:	HardwareStatusPageOffset[11:2]U32									
Value	Name									
[16, 1023]										
	1:0	Reserved <table border="1"><tr><td>Format:</td><td>MBZ</td></tr></table>	Format:	MBZ						
Format:	MBZ									
2	31:0	Data DWord 0 <table border="1"><tr><td>Format:</td><td>U32</td></tr></table> <p>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).</p>	Format:	U32						
Format:	U32									
3	31:0	Data DWord 1 <table border="1"><tr><td>Format:</td><td>U32</td></tr></table> <p>This field specifies the upper DWord value to be written to the targeted QWord location (DW 1).</p>	Format:	U32						
Format:	U32									



MI_STORE_DATA_INDEX

MI_STORE_DATA_INDEX			
Source:	VideoCS		
Length Bias:	2		
<p>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).</p>			
Programming Notes			
<ul style="list-style-type: none"> 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 Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	21h MI_STORE_DATA_INDEX
		Format:	OpCode
22	Reserved		
Format:	MBZ		
21	Use Per-Process Hardware Status Page If this bit is set, this command will index into the per-process hardware status page at offset 0K from the LRCA. If clear, the Global Hardware Status Page will be indexed. This bit must be '0' if the Execlist Enable bit is clear.		
20:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1) = 2 for QWord	
	Format:	=n Total Length - 2	
1	31:12	Reserved	
		Format:	MBZ



MI_STORE_DATA_INDEX										
	11:2	Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>U10 zero-based DWord offset into the HW status page</td> </tr> <tr> <td>Format:</td> <td>GraphicsAddress[11:2]U32</td> </tr> </table> <p>This field specifies the offset (into the hardware status page) to which the data will be written. For a QWord write, the offset is valid down to bit 3 only.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 60%;">Value</th> <th style="width: 40%;">Name</th> </tr> </thead> <tbody> <tr> <td>[16, 1023]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>The first few DWords of this status page are reserved for special-purpose data storage - targeting these reserved locations via this command is UNDEFINED.</p>	Format:	U10 zero-based DWord offset into the HW status page	Format:	GraphicsAddress[11:2]U32	Value	Name	[16, 1023]	
		Format:	U10 zero-based DWord offset into the HW status page							
		Format:	GraphicsAddress[11:2]U32							
		Value	Name							
		[16, 1023]								
1:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ								
2	31:0	Data DWord 0 <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>U32 FormatDesc</td> </tr> </table> <p>This field specifies the upper DWord value to be written to the targeted QWord location (DW 1).</p>	Format:	U32 FormatDesc						
Format:		U32 FormatDesc								
3	Data Word 1 <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>U32 FormatDesc</td> </tr> </table> <p>This field specifies the upper DWord value to be written to the targeted QWord location (DW 1).</p>	Format:	U32 FormatDesc							
Format:	U32 FormatDesc									



MI_STORE_REGISTER_MEM

MI_STORE_REGISTER_MEM			
Source:	CommandStreamer		
Length Bias:	2		
<p>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.</p>			
Programming Notes			
<ul style="list-style-type: none"> The command temporarily halts command execution. The memory address for the write is snooped on the host bus. This command should not be used from within a "non-privilege" batch buffer to access global virtual space. doing so will be treated as privilege access violation. Refer "User Mode Privilege Command" in MI_BATCH_BUFFER_START command section to know HW behavior on encountering privilege access violation. This command can be used within ring buffers and/or "privilege" batch buffers to access global virtual space. This command will cause undefined data to be written to memory if given register addresses for the PGTBL_CTL_0 or FENCE registers. 			
Workaround			
<p>Workaround: Reads to MMIO registers outside GT (Register Address > 0x40000) are not supported.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
28:23		MI Command Opcode	
		Default Value:	24h MI_STORE_REGISTER_MEM
		Format:	OpCode
22		Use Global GTT	
		Format:	Boolean
<p>It is allowed for this bit to be set when executing this command from a privileged (secure) batch or ring buffer. This bit must be clear when programmed from within a non-privileged batch buffer. This bit must be 1 if the Per Process GTT Enable bit is clear. This command will use the global GTT to translate the Address and this command must be executing from a privileged (secure) batch buffer.</p>			



MI_STORE_REGISTER_MEM							
	21	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Source:</td> <td>BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS	Format:	MBZ	
	Source:	BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS					
	Format:	MBZ					
	21	Predicate Enable <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Source:</td> <td>RenderCS</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>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.</p>	Source:	RenderCS	Format:	U1	
	Source:	RenderCS					
	Format:	U1					
20	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
19	Reserved						
18:8	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td style="width: 35%;">Format:</td> <td>=n Total Length - 2</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>2h</td> <td>Excludes DWord (0,1) [Default]</td> </tr> </tbody> </table>	Format:	=n Total Length - 2	Value	Name	2h	Excludes DWord (0,1) [Default]
Format:	=n Total Length - 2						
Value	Name						
2h	Excludes DWord (0,1) [Default]						
1	31:23	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ					
	22:2	Register Address <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MMIOAddress[22:2]MMIO_Register</td> </tr> </table> <p>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.</p> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="text-align: center; color: blue;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> Storing a VGA register is not permitted and will store an UNDEFINED value. The values of PGTBL_CTL0 or any of the FENCE registers cannot be stored to memory; UNDEFINED values will be written to memory if the addresses of these registers are specified. </td> </tr> </tbody> </table>	Format:	MMIOAddress[22:2]MMIO_Register	Programming Notes	<ul style="list-style-type: none"> Storing a VGA register is not permitted and will store an UNDEFINED value. The values of PGTBL_CTL0 or any of the FENCE registers cannot be stored to memory; UNDEFINED values will be written to memory if the addresses of these registers are specified. 	
Format:	MMIOAddress[22:2]MMIO_Register						
Programming Notes							
<ul style="list-style-type: none"> Storing a VGA register is not permitted and will store an UNDEFINED value. The values of PGTBL_CTL0 or any of the FENCE registers cannot be stored to memory; UNDEFINED values will be written to memory if the addresses of these registers are specified. 							
1:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						



MI_STORE_REGISTER_MEM				
2..3	63:2	<p>Memory Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>GraphicsAddress[63:2]MMIO</td> </tr> </table> <p>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[63:2] for a DWord register GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p>	Format:	GraphicsAddress[63:2]MMIO
	Format:	GraphicsAddress[63:2]MMIO		
1:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			



MI_STORE_URB_MEM

MI_STORE_URB_MEM		
Source:	RenderCS	
Length Bias:	2	
<p>The MI_STORE_URB_MEM command requests a URB read from a specified memory mapped URB location in the device and store of that DWord to memory. The URB address is specified along with the command to perform the read.</p>		
Programming Notes		
<ul style="list-style-type: none"> The command temporarily halts command execution 		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 0h MI_COMMAND
		Format: OpCode
	28:23	MI Command Opcode
		Default Value: 2Dh MI_STORE_URB_MEM
22:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Format: =n	
	Total Length - 2. Excludes DWord (0,1).	
	Value	Name
	2h	[Default]
1	31:15	Reserved
		Format: MBZ
	14:2	URB Address This field specifies Bits 14:2 of the URB offset the DWord will be read in the URB. This command only supports reading from the lower 32KB of the URB space.
1:0	Reserved	
	Format: MBZ	



MI_STORE_URB_MEM				
2..3	63:6	<p>Memory Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[63:6]</td> </tr> </table> <p>This field specifies the address of the location of where the value will be written to memory. The value must be in the first DW location of the cache line. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p>	Format:	GraphicsAddress[63:6]
	Format:	GraphicsAddress[63:6]		
5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			



MI_SUSPEND_FLUSH

MI_SUSPEND_FLUSH		
Source:	VideoEnhancementCS	
Length Bias:	1	
Description		
Blocks PM Flush Requests.		
DWord	Bit	Description
0	31:29	Command Type Default Value: 0h MI_COMMAND
	28:23	MI Command Opcode Default Value: 0Bh MI_SUSPEND_FLUSH
	22:1	Reserved Format: MBZ
	0	Suspend Flush Format: Enable Description This field suspends flush due to a PM flush request.



MI_SUSPEND_FLUSH

MI_SUSPEND_FLUSH		
Source:	BlitterCS	
Length Bias:	1	
Description		
Blocks PM Flush Requests.		
DWord	Bit	Description
0	31:29	Command Type Default Value: 0h MI_COMMAND
	28:23	MI Command Opcode Default Value: 0Bh MI_SUSPEND_FLUSH
	22:1	Reserved Format: MBZ
	0	Suspend Flush Format: Enable <div style="text-align: center;">Description</div> This field suspends flush due to a PM flush request.



MI_SUSPEND_FLUSH

MI_SUSPEND_FLUSH				
Source:	RenderCS			
Length Bias:	1			
Description				
Blocks PM Flush Requests.				
DWord	Bit	Description		
0	31:29	Command Type		
		Default Value:	0h MI_COMMAND	
		Format:	OpCode	
	28:23	MI Command Opcode		
		Default Value:	0Bh MI_SUSPEND_FLUSH	
		Format:	OpCode	
	22:1	Reserved		
		Format:	MBZ	
	0	Suspend Flush	Format:	Enable
			Description	
		This field suspends flush due to a PM flush request.		



MI_SUSPEND_FLUSH

MI_SUSPEND_FLUSH		
Source:	VideoCS	
Length Bias:	1	
Description		
Blocks PM Flush Requests.		
DWord	Bit	Description
0	31:29	Command Type Default Value: 0h MI_COMMAND
	28:23	MI Command Opcode Default Value: 0Bh MI_SUSPEND_FLUSH
	22:1	Reserved Format: MBZ
	0	Suspend Flush Format: Enable <div style="text-align: center;">Description</div> This field suspends flush due to a PM flush request.



MI_TOPOLOGY_FILTER

MI_TOPOLOGY_FILTER			
Source:	RenderCS		
Length Bias:	1		
<p>This command is used to specify a specific 3DPrimType value, where the CS will ignore all 3DPRIMITIVE commands that do not have a matching 3DPrimType. This primitive culling is optional (turned off by using this command with a Topology Filter Value of 0). This command is specific to the Render command stream only.</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	0Dh MI_TOPOLOGY_FILTER
		Format:	OpCode
	22:6	Reserved	
		Format:	MBZ
	5:0	Topology Filter Value	
		Format:	3D_Prim_Topo_Type
		When non-zero, the CS will discard all 3DPRIMITIVE commands which do not match the specified 3DPrimTopologyType. When zero, no filtering is performed (normal operation).	



MI_UPDATE_GTT

MI_UPDATE_GTT		
Source:	BSpec	
Length Bias:	2	
DWord	Bit	Description
		<p>The MI_UPDATE_GTT command is used to update GTT page table entries in a coherent manner and at a predictable place in the command flow.</p> <p>A PIPE_CONTROL flush command with "CS Stall" bit set must be programmed prior to MI_UPDATE_GTT command, since work associated with preceding commands that are still in the pipeline may be referencing GTT entries that will be changed by its execution. The flush must also invalidate TLBs and read caches that may become invalid as a result of the changed GTT entries. A PIPE_CONTROL flush command with "CS Stall" bit set must be programmed post MI_UPDATE_GTT command to ensure the GGTT is updated with modified page table entries before the following workload references the modified entries.</p> <p>PIPE_CONTROL flush is not required if it can be guaranteed that the pipeline is free of any work that relies on changing GTT entries (such as MI_UPDATE_GTT contained in a paging DMA buffer that is doing only update/mapping activities and no rendering).</p> <p>MI_UPDATE_GTT command is privilege operation and will be converted to a no-op and an error flagged if it is executed from within a non-secure batch buffer.</p> <p>PPGTT updates cannot be done via MI_UPDATE_GTT, gfx driver will have to use MI_STORE_DATA_IMM for PPGTT inline updates.</p>
		<p>The MI_UPDATE_GTT command is used to update GGTT page table entries in a coherent manner and at a predictable place in the command flow. A MI_FLUSH_DWORD flush command with "CS Stall" bit set must be programmed prior to MI_UPDATE_GTT command, since work associated with preceding commands that are still in the pipeline may be referencing GTT entries that will be changed by its execution. The flush must also invalidate TLBs and read caches that may become invalid as a result of the changed GTT entries. A MI_FLUSH_DWORD flush command with "CS Stall" bit set must be programmed post MI_UPDATE_GTT command to ensure the GGTT is updated with modified page table entries before the following workload references the modified entries. MI_FLUSH_DWORD flush is not required if it can be guaranteed that the pipeline is free of any work that relies on changing GTT entries (such as MI_UPDATE_GTT contained in a paging DMA buffer that is doing only update/mapping activities and no rendering).</p> <p>MI_UPDATE_GTT command is privilege operation and will be converted to a no-op and an error flagged if it is executed from within a non-secure batch buffer. PPGTT updates cannot be done via MI_UPDATE_GTT, gfx driver will have to use MI_STORE_DATA_IMM for PPGTT inline updates.</p>
		<p>RenderCS</p>
		<p>BlitterCS, VideoCS, VideoEnhancementCS</p>



MI_UPDATE_GTT						
0	31:29	Command Type <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>0h MI_COMMAND</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	0h MI_COMMAND	Format:	OpCode
	Default Value:	0h MI_COMMAND				
	Format:	OpCode				
	28:23	MI Command Opcode <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>23h MI_UPDATE_GTT</td> </tr> <tr> <td>Format:</td> <td>OpCode</td> </tr> </table>	Default Value:	23h MI_UPDATE_GTT	Format:	OpCode
Default Value:	23h MI_UPDATE_GTT					
Format:	OpCode					
22:10	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
9:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>=n Total Length - 2. Excludes DWord (0,1).</td> </tr> </table>	Default Value:	0h	Format:	=n Total Length - 2. Excludes DWord (0,1).	
Default Value:	0h					
Format:	=n Total Length - 2. Excludes DWord (0,1).					
1	31:12	Entry Address <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>This field holds the QW offset of the first table entry to be modified in GGTT.</p>	Format:	GraphicsAddress[31:12]		
	Format:	GraphicsAddress[31:12]				
11:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
2..n	63:0	Entry Data <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>PageTableEntry</td> </tr> </table> <p>This Dword becomes the new page table entry. See PPGTT/Global GTT Table Entries (PTEs) in Memory Interface Registers.</p>	Format:	PageTableEntry		
Format:	PageTableEntry					



MI_URB_ATOMIC_ALLOC

MI_URB_ATOMIC_ALLOC						
Source:	RenderCS					
Length Bias:	1					
This command is used to specify the region in URB allocated for URB atomic value storage. This command is specific to the Render command stream only.						
Programming Notes						
This command can only be sent after a flush has occurred.						
DWord	Bit	Description				
0	31:29	Command Type				
		Default Value: 0h MI_COMMAND				
	Format: OpCode					
	28:23	MI Command Opcode				
		Default Value: 09h MI_URB_ALLOC				
	Format: OpCode					
	22:20	Reserved				
		Format: MBZ				
	19:12	URB Atomic Storage Offset				
		Format: U8 Number of 128B Entries				
This field specifies the offset of a 128B granular starting address in the URB. The value of URB Atomic Storage Offset plus the value of the URB Atomic Storage Size must not exceed 256.						
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,255]</td> <td></td> <td>0-(32KB-128B)</td> </tr> </tbody> </table>		Value	Name	Description	[0,255]	
Value	Name	Description				
[0,255]		0-(32KB-128B)				
11:9	Reserved					
	Format: MBZ					
8:0	URB Atomic Storage Size					
	Format: U9 Number of 128B Entries					
	This field specifies the size of the buffer in the URB in number of 128B entries. If this field has a value of zero then the URB Atomic allocation is disabled and will not be context save/restored.					
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,256]</td> <td></td> <td>0-32KB</td> </tr> </tbody> </table>	Value	Name	Description	[0,256]	
Value	Name	Description				
[0,256]		0-32KB				



MI_USER_INTERRUPT

MI_USER_INTERRUPT			
Source:	VideoEnhancementCS		
Length Bias:	1		
The MI_USER_INTERRUPT command is used to generate a User Interrupt condition. The parser will continue parsing after processing this command. See User Interrupt.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	02h MI_USER_INTERRUPT
		Format:	OpCode
	22:0	Reserved	
		Format:	MBZ



MI_USER_INTERRUPT

MI_USER_INTERRUPT		
Source:	BlitterCS	
Length Bias:	1	
<p>The MI_USER_INTERRUPT command is used to generate a User Interrupt condition. The parser will continue parsing after processing this command. See User Interrupt.</p>		
DWord	Bit	Description
0	31:29	Command Type Default Value: 0h MI_COMMAND
	28:23	MI Command Opcode Default Value: 02h MI_USER_INTERRUPT
	22:0	Reserved Format: MBZ



MI_USER_INTERRUPT

MI_USER_INTERRUPT			
Source:	RenderCS		
Length Bias:	1		
The MI_USER_INTERRUPT command is used to generate a User Interrupt condition. The parser will continue parsing after processing this command. See User Interrupt.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	02h MI_USER_INTERRUPT
		Format:	OpCode
	22:0	Reserved	
		Format:	MBZ



MI_USER_INTERRUPT

MI_USER_INTERRUPT		
Source:	VideoCS	
Length Bias:	1	
<p>The MI_USER_INTERRUPT command is used to generate a User Interrupt condition. The parser will continue parsing after processing this command. See User Interrupt.</p>		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 0h MI_COMMAND
		Format: OpCode
	28:23	MI Command Opcode
		Default Value: 02h MI_USER_INTERRUPT
		Format: OpCode
	22:0	Reserved
		Format: MBZ



MI_WAIT_FOR_EVENT

MI_WAIT_FOR_EVENT

Source: RenderCS, BlitterCS
 Length Bias: 1

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).

Execution List Mode of Scheduling: CS on evaluating MI_WAIT_FOR_EVENT to be unsuccessful (has to wait for event to happen) triggers synchronous context switch stating the switch reason in Context Status Buffer. Note that synchronous context switch can be inhibited through programming "Inhibit Synchronous Context Switch" bit in CTXT_SR_CTL register or by disabling arbitration through MI_ARB_ON_OFF command.

DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	0h MI_COMMAND
		Format:	OpCode
	28:23	MI Command Opcode	
		Default Value:	03h MI_WAIT_FOR_EVENT
		Format:	OpCode
22	Reserved		
	Format:	MBZ	



MI_WAIT_FOR_EVENT

21	<p>Display Plane 1 C Vertical Blank Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait until the next Display Plane 1 C "Vertical Blank" event occurs. This event is described as the start of the next Display C vertical blank period. Note that this can cause a wait for up to an entire refresh period. See Vertical Blank Event in the Device Programming Interface chapter of MI Functions.</p>	Format:	Enable
Format:	Enable		
20	<p>Display Plane 6 Flip Pending Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait for the duration of a Display Plane 2 C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers).</p>	Format:	Enable
Format:	Enable		
19	<p>Display Plane 12 Flip Pending Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait for the duration of a Display Plane 4 C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers).</p>	Format:	Enable
Format:	Enable		
18	<p>Display Plane 11 Flip Pending Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait for the duration of a Display Plane 4 B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers).</p>	Format:	Enable
Format:	Enable		
17	<p>Display Plane 10 Flip Pending Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait for the duration of a Display Plane 4 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).</p>	Format:	Enable
Format:	Enable		

**MI_WAIT_FOR_EVENT**

	16	Display Plane 9 Flip Pending Wait Enable Format: <table border="1"><tr><td></td><td>Enable</td></tr></table> <p>This field enables a wait for the duration of a Display Plane 3 C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers).</p>		Enable
		Enable		
	15	Display Plane 3 Flip Pending Wait Enable Format: <table border="1"><tr><td></td><td>Enable</td></tr></table> <p>This field enables a wait for the duration of a Display Plane 1 C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers).</p>		Enable
		Enable		
	14	Display Plane 1 C Scan Line Wait Enable Format: <table border="1"><tr><td></td><td>Enable</td></tr></table> <p>This field enables a wait while a Display Plane 1 C "Scan Line" condition exists. This condition is defined as the the start of the scan line specified in the Pipe C Display Scan Line Count Range Compare Register.</p>		Enable
		Enable		
13:12	Reserved Format: <table border="1"><tr><td></td><td>MBZ</td></tr></table>		MBZ	
	MBZ			
11	Display Plane 1 B Vertical Blank Wait Enable Format: <table border="1"><tr><td></td><td>Enable</td></tr></table> <p>This field enables a wait until the next Display Plane 1 B "Vertical Blank" event occurs. This event is described as the start of the next Display B vertical blank period. Note that this can cause a wait for up to an entire refresh period.</p>		Enable	
	Enable			
10	Display Plane 5 Flip Pending Wait Enable Format: <table border="1"><tr><td></td><td>Enable</td></tr></table> <p>This field enables a wait for the duration of a Display Plane 2 B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers).</p>		Enable	
	Enable			



MI_WAIT_FOR_EVENT			
9	<p>Display Plane 2 Flip Pending Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait for the duration of a Display Plane B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers).</p>	Format:	Enable
Format:	Enable		
8	<p>Display Plane 1 B Scan Line Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait while a Display Plane 1 B "Scan Line" condition exists. This condition is defined as the the start of the scan line specified in the Pipe B Display Scan Line Count Range Compare Register.</p>	Format:	Enable
Format:	Enable		
7	<p>Display Plane 8 Flip Pending Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait for the duration of a Display Plane 3 B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers).</p>	Format:	Enable
Format:	Enable		
6	<p>Display Plane 7 Flip Pending Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait for the duration of a Display Plane 3 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).</p>	Format:	Enable
Format:	Enable		
5:4	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
3	<p>Display Plane 1 A Vertical Blank Wait Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables a wait until the next Display Plane 1 A "Vertical Blank" event occurs. This event is described as the start of the next Display A vertical blank period. Note that this can cause a wait for up to an entire refresh period.</p>	Format:	Enable
Format:	Enable		



MI_WAIT_FOR_EVENT			
2	Display Plane 4 Flip Pending Wait Enable <table border="1"><tr><td>Format:</td><td>Enable</td></tr></table> <p>This field enables a wait for the duration of a Display Plane 2 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).</p>	Format:	Enable
Format:	Enable		
1	Display Plane 1 Flip Pending Wait Enable <table border="1"><tr><td>Format:</td><td>Enable</td></tr></table> <p>This field enables a wait for the duration of a Display Plane 1 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).</p>	Format:	Enable
Format:	Enable		
0	Display Plnae 1 A Scan Line Wait Enable <table border="1"><tr><td>Format:</td><td>Enable</td></tr></table> <p>This field enables a wait while a Display Plane 1 A "Scan Line" condition exists. This condition is defined as the the start of the scan line specified in the Pipe A Display Scan Line Count Range Compare Register.</p>	Format:	Enable
Format:	Enable		



Move

mov - Move			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>A mov with the same source and destination type, no source modifier, and no saturation is a raw move. A packed byte destination region (B or UB type with HorzStride == 1 and ExecSize > 1) can only be written using raw move.</p> <p>When denorm mode is flush to zero, a raw mov instruction with saturation modifier will not flush the denorm input or output to zero (Denorm is preserved).</p> <p>Format: [(pred)] mov[.cmod] (exec_size) dst src0</p>			
Programming Notes			
A mov instruction with a source modifier always copies a denorm source value to a denorm destination value (in the manner of a raw move).			
There is no direct conversion from B/UB to DF or DF to B/UB. Use two instructions and a word or DWord intermediate type.			
There is no direct conversion from B/UB to Q/UQ or Q/UQ to B/UB. Use two instructions and a word or DWord intermediate integer type.			
There is no direct conversion from HF to DF or DF to HF. Use two instructions and F (Float) as an intermediate type.			
There is no direct conversion from HF to Q/UQ or Q/UQ to HF. Use two instructions and F (Float) or a word integer type or a DWord integer type as an intermediate type.			
Restriction			
Restriction : Raw move is not supported for Float values in ALT mode if any values are infinities or NaNs.			
Restriction : An accumulator can be a source or destination operand but not both.			
Syntax			
<pre>[(pred)] mov[.cmod] (exec_size) reg reg [(pred)] mov[.cmod] (exec_size) reg imm32 [(pred)] mov[.cmod] (exec_size) reg imm64</pre>			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src0.chan[n]; } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y



mov - Move		
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
DF		DF
*W,*D		*W,*D
*W,*D		*Q
*Q		*W,*D
*Q		*Q
F		*Q
DF		*Q
*Q		F
*Q		DF
*B,*W,*D		HF
F		HF
HF		*B,*W,*D
HF		F
HF		HF

DWord	Bit	Description
0..3	127:64	RegSource
		Exists If: (([Operand Controls][Src0.RegFile]!='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource
Exists If: (([Operand Controls][Src0.RegFile]=='IMM')		
Format: EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls	
	Format: EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header	
	Format: EU_INSTRUCTION_HEADER	



Move Indexed

movi - Move Indexed	
Source:	Eulsa
Length Bias:	4
<p>The movi instruction performs a fast component-wise indexed move for subfields from src0 to dst. The source operand must be an indirectly-addressed register. All channels of the source operand share the same register number, 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.</p> <p>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 eight subfields can be selected by an instruction.</p>	
Format: [(pred)] movi (exec_size) dst src0	
Programming Notes	
<p>HW Implementation Details:</p> <p>The source register is calculated by adding the register portion of the first index register with the register portion of the address immediate, $a0.0[11:5] + \text{addr_imm}[9:5]$</p> <p>For byte movi, byte0 of the destination is selected by $(a0.1[4:0])$, byte1 is selected by $(a0.1[4:0])$, ..., and byte7 is selected by $(a0.7[4:0])$. The rest of the bytes are undefined.</p> <p>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.</p> <p>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 $(a0.7[4:2] \& 11b)$.</p> <p>For all 3 conditions above, $a0.n[4:0] = a0.n[4:0] + \text{addr_imm}[4:0]$.</p>	
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 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.	



movi - Move Indexed

Restriction : The destination offset is only used to create channel enables. Each element of the 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 :

Only 8 address subregisters are used (a0.0-a0.7). Destination element 8 will be sourced from address register zero (a0.0), dst.9 <-a0.1, etc. This is an exception to the above restriction, for example:

movi (8) r31.8:uw r[a0.0,0]<8;8,1>:uw // r31.8:uw<-a0.0:uw, r31.9:uw<-a0.1:uw, 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] ) { srcreg
= subreg(address[n] + addr_imm); dst.chan[n] = srcregfile.srcreg.srcreg; } }
```

Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	Y	Y

Src Types	Dst Types
B	B
UB	UB
W	W
UW	UW
D	D
UD	UD
F	F

DWord	Bit	Description
0..3	127:64	RegSource
		Exists If: ([Operand Controls][Src0.RegFile]!='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource
Exists If: ([Operand Controls][Src0.RegFile]=='IMM')		
Format: EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls	
Format: EU_INSTRUCTION_OPERAND_CONTROLS		
31:0	Header	
	Format: EU_INSTRUCTION_HEADER	



Multiply

mul - Multiply			
Source:	Eulsa		
Length Bias:	4		
<p>The mul instruction performs component-wise multiplication of src0 and src1 and stores the results in dst. When multiplying integer datatypes, if src0 is DW and src1 is W, irrespective of the destination datatype, the accumulator maintains full 48-bit precision. This is required to handle the macro for 32x32 multiplication. The macro described in the mach instruction should be used to obtain the full precision 64-bit multiplication results.</p> <p>Note: A 32x32 multiply operation is handled natively, without a macro. When operating in this mode, the resulting 64-bit data is packed, unlike the macro, where the lower and upper 32 bits of the result are written to different general registers by two separate instructions. Refer to the macro description for details.</p> <p>When multiplying integer data types, if one of the sources is a DW, the resulting full precision data is stored in the accumulator. However, if the destination data type is either W or DW, the low bits of the result are written to the destination register and the remaining high bits are discarded. This results in undefined Overflow and Sign flags. Therefore, conditional modifiers and saturation (.sat) cannot be used in this case.</p>			
Format: [(pred)] mul[.cmod] (exec_size) dst src0 src1			
Restriction			
Restriction : Integer source operands cannot be accumulators.			
Restriction : When multiplying a DW and any lower precision integer, the DW operand must on src0.			
Syntax			
[(pred)] mul[.cmod] (exec_size) reg reg reg [(pred)] mul[.cmod] (exec_size) reg reg imm32			
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	Saturation	Source Modifier
Y	Y	Y	Y
Src Types		Dst Types	
*B		*B	
*B		*W	
*B		*D	



mul - Multiply		
*W		*W
*W		*D
*D		*D
*D		*Q
F		F
DF		DF
HF		HF
HF, F		HF, F
DWord	Bit	Description
0..3	127:64	RegSource
		Exists If: ([RegSource][Src1.RegFile]!='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile]='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
31:0	Header	
	Format: EU_INSTRUCTION_HEADER	



Multiply Accumulate

mac - Multiply Accumulate			
Source:	Eulsa		
Length Bias:	4		
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)] mac[.cmod] (exec_size) dst src0 src1		
Programming Notes			
When source and destination datatypes are different, the implied datatype for the accumulator operand is always the destination datatype.			
Restriction			
Restriction : Accumulator is an implicit source and thus cannot be an explicit source operand.			
Syntax			
[(pred)] mac[.cmod] (exec_size) reg reg reg [(pred)] mac[.cmod] (exec_size) reg reg imm32			
Pseudocode			
<pre> Evaluate (WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { dst.chan[n] = src0.chan[n] * src1.chan[n] + acc0.chan[n]; } } </pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types		Dst Types	
*B,*W		*B,*W,*D	
F		F	
DF		DF	
HF		HF	
HF, F		HF, F	
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG



mac - Multiply Accumulate		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Multiply Accumulate High

mach - Multiply Accumulate High	
Source:	Eulsa
Length Bias:	4
<p>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 src0 with the high word of the DWord in src1, 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 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.</p> <pre>mul (8) acc0:d r2.0<8;8,1>:d r3.0<16;8,2>:uw mach (8) r5.0<1>:d r2.0<8;8,1>:d r3.0<8;8,1>:d mov (8) r6.0<1>:d acc0:d // Low 32 bits.</pre> <p>Here is a different example including negation. An added preliminary mov is required for source modification on src1.</p> <pre>mov (8) r3.0<1>:d -r3<8;8,1>:d mul (8) acc0:d r2.0<8;8,1>:d r3.0<16;8,2>:uw mach (8) r5.0<1>:d r2.0<8;8,1>:d r3.0<8;8,1>:d // High 32 bits mov (8) r6.0<1>:d acc0:d // Low 32 bits.</pre> <p>The mach should have channel enable from the destHI of IMUL, the 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.</p>	
<p>Format:</p> <pre>[(pred)] mach[.cmod] (exec_size) dst src0 src1</pre>	
Restriction	
Restriction : Accumulator is an implicit source and thus cannot be an explicit source operand.	
Restriction : AccWrEn is required.	
Syntax	
<pre>[(pred)] mach[.cmod] (exec_size) reg reg reg [(pred)] mach[.cmod] (exec_size) reg reg imm32</pre>	
Pseudocode	



mach - Multiply Accumulate High

```

Evaluate(WrEn);
for ( n = 0; n < exec_size; n++ ) {
    if ( WrEn.chan[n] ) {
        acc.chan[n][63:0] = (src1.chan[n][31:16] *
            src0.chan[n][31:0]) << 16 + acc.chan[n][63:0];
        dst.chan[n][31:0] = acc.chan[n][63:32];
    }
}
    
```

Errata	Description
	A source modifier must not be used on src1 for the macro operation. This applies to both mul and mach of the macro. If source modifier is required, an additional mov instruction may be used before the macro.

Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	Y	Y

Src Types	Dst Types
D	D
UD	UD

DWord	Bit	Description
0..3	127:64	RegSource
		Exists If: ([RegSource][Src1.RegFile] != 'IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource
Exists If: ([ImmSource][Src1.RegFile] == 'IMM')		
Format: EU_INSTRUCTION_SOURCES_REG_IMM		
63:32	Operand Controls	
	Format: EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header	
	Format: EU_INSTRUCTION_HEADER	



Multiply Add

mad - Multiply Add			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>The conditional modifier and saturation (.sat) must not be used when src1 or src2 are dwords.</p>			
Format:	[(pred)] mad[.cmod] (exec_size) dst src0 src1 src2		
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 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	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types		Dst Types	
F		F	
DF		DF	
HF		HF	
HF, F		HF, F	
DWord	Bit	Description	
0..3	127:126	Reserved	
		Format:	MBZ
	125:106	Source 2	
		Format:	EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC



mad - Multiply Add												
105	Reserved	Format: MBZ										
104:85	Source 1	Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC										
84	Reserved	Format: MBZ										
83:64	Source 0	Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC										
63:56	Destination Register Number	Format: DstRegNum										
55:53	Destination Subregister Number	Format: DstSubRegNum[2:0]										
52:49	Destination Channel Enable	Format: ChanEn[4] <p>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 x, y, z, and w, respectively, where x corresponds to Channel 0 in the group and w corresponds to channel 3 in the group</p>										
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 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Single Precision Float</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>DWord</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Unsigned DWord</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Double Precision Float</td> </tr> </tbody> </table>	Value	Name	00b	Single Precision Float	01b	DWord	10b	Unsigned DWord	11b	Double Precision Float
Value	Name											
00b	Single Precision Float											
01b	DWord											
10b	Unsigned DWord											
11b	Double Precision Float											
48:42	Reserved	Format: MBZ										



mad - Multiply Add		
43:42	Source Data Type This field contains the data type for all three sources	
	Value	Name
	00b	Single Precision Float
	01b	DWord
	10b	Unsigned DWord
	11b	Double Precision Float
41:40	Source 2 Modifier	
	Exists If:	/// ([Property[Source Modifier]='true')
	Format:	SrcMod
39:38	Source 1 Modifier	
	Exists If:	/// ([Property[Source Modifier]='true')
	Format:	SrcMod
41:36	Reserved	
	Exists If:	/// ([Property[Source Modifier]='false')
	Format:	MBZ
37:36	Source 0 Modifier	
	Exists If:	/// ([Property[Source Modifier]='true')
	Format:	SrcMod
35	Reserved	
	Format:	MBZ
34	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	Destination Register File	
	Value	Name
	0	GRF
	1	MRF
32	Reserved	
	Format:	MBZ



mad - Multiply Add		
	31:0	Header
	Format:	EU_INSTRUCTION_HEADER



Multiply Add for Macro

madm - Multiply Add for Macro			
Source:	Eulsa		
Length Bias:	4		
<p>The madm 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.</p> <p>The source and destination operands have a higher precision carried in the exponent for this operation. The madm instruction is used for macro operations, where precision is accumulated over several instructions. This accumulation requires the exponent to increase by 2 extra bits across multiple madm operations.</p> <p>Refer to Macros Defined in 'Math' Section for usage and restrictions of this operation.</p>			
<p>Format:</p> <p>[(pred)] madm[.cmod] (exec_size) dst src0 src1 src2</p>			
Restriction			
<p>Restriction : Accumulator access is restricted to the special accumulators (acc2-acc9). Refer to the Accumulator Section for details on the special accumulators.</p>			
Syntax			
<p>[(pred)] madm[.cmod] (exec_size) reg reg reg reg</p>			
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	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
F	F		
DF	DF		
DWord	Bit	Description	
0..3	127:126	Reserved	
		Format:	MBZ
	125:106	Source 2	
		Format:	EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC
105		Reserved	
		Format:	MBZ
104:85		Source 1	
		Format:	EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC



madm - Multiply Add for Macro												
84	Reserved	Format: MBZ										
83:64	Source 0	Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC										
63:56	Destination Register Number	Format: DstRegNum										
55:53	Destination Subregister 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 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 Type This field contains the data type for the destination	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Single Precision Float</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>DWord</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Unsigned DWord</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Double Precision Float</td> </tr> </tbody> </table>	Value	Name	00b	Single Precision Float	01b	DWord	10b	Unsigned DWord	11b	Double Precision Float
Value	Name											
00b	Single Precision Float											
01b	DWord											
10b	Unsigned DWord											
11b	Double Precision Float											
48:42	Reserved	Format: MBZ										
43:42	Source Data Type This field contains the data type for all three sources	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Single Precision Float</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>DWord</td> </tr> </tbody> </table>	Value	Name	00b	Single Precision Float	01b	DWord				
Value	Name											
00b	Single Precision Float											
01b	DWord											



madm - Multiply Add for Macro		
	10b	Unsigned DWord
	11b	Double Precision Float
41:40	Source 2 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
39:38	Source 1 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
41:36	Reserved	
	Exists If:	/// ([Property[Source Modifier] == 'false')
	Format:	MBZ
37:36	Source 0 Modifier	
	Exists If:	/// ([Property[Source Modifier] == 'true')
	Format:	SrcMod
35	Reserved	
	Format:	MBZ
34	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	Destination Register File	
	Value	Name
	0	GRF
	1	MRF
32	Reserved	
	Format:	MBZ
31:0	Header	
	Format:	EU_INSTRUCTION_HEADER



No Operation

nop - No Operation									
Source:	Eulsa								
Length Bias:	4								
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.									
Syntax									
nop									
Pseudocode									
{ ; // The null statement, which does nothing. }									
Predication	Conditional Modifier	Saturation	Source Modifier						
N	N	N	N						
DWord	Bit	Description							
0..3	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. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No Breakpoint [Default]</td> </tr> <tr> <td>1</td> <td>Breakpoint</td> </tr> </tbody> </table>		Value	Name	0	No Breakpoint [Default]	1	Breakpoint
	Value	Name							
	0	No Breakpoint [Default]							
1	Breakpoint								
29:7	Reserved Format: _____ MBZ								
6:0	Opcode Format: _____ EU_OPCODE								



Oword Aligned Block Read MSD

MSD0R_OWAB - Oword Aligned Block Read MSD		
Source:	DataPort 0	
Length Bias:	1	
Family:	Block R/W	
Group:	OW Aligned Block R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHR Indicates that the message requires a header.
	18	Legacy Message
		Default Value: 0h Format: Opcode Legacy Message
	17:14	Message Type
		Default Value: 01h Format: Opcode Aligned Block Read message
	13	Reserved
Format: MBZ Ignored		
12:11	Reserved	
	Format: MBZ Ignored	
10:8	Data Elements	
	Format: MDC_DB_OW Specifies the number of contiguous Owords to be read	
7:0	Binding Table Index	
	Format: MDC_BTS_A32 Specifies the Binding Table Index for the message	



Oword Block Read MSD

MSDOR_OWB - Oword Block Read MSD						
Source:	DataPort 0					
Length Bias:	1					
Family:	Block R/W					
Group:	OW Block R/W					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_MHR</td> </tr> </table> Indicates that the message requires a header.	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18	Legacy Message <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">0h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Legacy Message	Default Value:	0h	Format:	Opcode
	Default Value:	0h				
	Format:	Opcode				
	17:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="text-align: center;">00h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Block Read message	Default Value:	00h	Format:	Opcode
	Default Value:	00h				
	Format:	Opcode				
13	Invalidate After Read <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_IAR</td> </tr> </table> Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs	Format:	MDC_IAR			
Format:	MDC_IAR					
12:11	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
10:8	Data Elements <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_DB_OW</td> </tr> </table> Specifies the number of contiguous Owords to be read or written	Format:	MDC_DB_OW			
Format:	MDC_DB_OW					
7:0	Binding Table Index <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDC_BTS_A32</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS_A32			
Format:	MDC_BTS_A32					



Oword Block Write MSD

MSD0W_OWB - Oword Block Write MSD						
Source:	DataPort 0					
Length Bias:	1					
Family:	Block R/W					
Group:	OW Block R/W					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_MHR</td> </tr> </table> Indicates that the message requires a header.	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18	Legacy Message <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%; text-align: center;">0h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Legacy Message	Default Value:	0h	Format:	Opcode
	Default Value:	0h				
	Format:	Opcode				
	17:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%; text-align: center;">08h</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">Opcode</td> </tr> </table> Block Write message	Default Value:	08h	Format:	Opcode
	Default Value:	08h				
Format:	Opcode					
13	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
12:11	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MBZ</td> </tr> </table> Ignored	Format:	MBZ			
Format:	MBZ					
10:8	Data Elements <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_DB_OW</td> </tr> </table> Specifies the number of contiguous Owords to be read or written	Format:	MDC_DB_OW			
Format:	MDC_DB_OW					
7:0	Binding Table Index <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%; text-align: center;">MDC_BTS_A32</td> </tr> </table> Specifies the Binding Table Index for the message	Format:	MDC_BTS_A32			
Format:	MDC_BTS_A32					



Oword Dual Block Read MSD

MSD0R_OWDB - Oword Dual Block Read MSD						
Source:	DataPort 0					
Length Bias:	1					
Family:	Block R/W					
Group:	OW Dual Block R/W					
DWord	Bit	Description				
0	19	Header Present <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	Enable		
	Format:	Enable				
	18	Legacy Message <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Legacy Message</p>	Default Value:	0h	Format:	Opcode
	Default Value:	0h				
	Format:	Opcode				
	17:14	Message Type <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>02h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Oword Block Read message</p>	Default Value:	02h	Format:	Opcode
	Default Value:	02h				
Format:	Opcode					
13	Invalidate After Read <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MDC_IAR</td> </tr> </table> <p>Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs</p>	Format:	MDC_IAR			
Format:	MDC_IAR					
12:10	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
9:8	Data Elements <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MDC_DB_OWD</td> </tr> </table> <p>Specifies the number of contiguous Owords to be read or written</p>	Format:	MDC_DB_OWD			
Format:	MDC_DB_OWD					
7:0	Binding Table Index <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MDC_BTS_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_A32			
Format:	MDC_BTS_A32					



Oword Dual Block Write MSD

MSD0W_OWDB - Oword Dual Block Write MSD						
Source:	DataPort 0					
Length Bias:	1					
Family:	Block R/W					
Group:	OW Dual Block R/W					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	Enable		
	Format:	Enable				
	18	<p>Legacy Message</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Legacy Message</p>	Default Value:	0h	Format:	Opcode
	Default Value:	0h				
	Format:	Opcode				
	17:14	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0Ah</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Oword Block Read message</p>	Default Value:	0Ah	Format:	Opcode
Default Value:	0Ah					
Format:	Opcode					
13:10	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
9:8	<p>Data Elements</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MDC_DB_OWD</td> </tr> </table> <p>Specifies the number of contiguous Owords to be read or written</p>	Format:	MDC_DB_OWD			
Format:	MDC_DB_OWD					
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MDC_BTS_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_A32			
Format:	MDC_BTS_A32					



PIPE_CONTROL

PIPE_CONTROL			
Source:	RenderCS		
Length Bias:	2		
The PIPE_CONTROL command is used to effect the synchronization described above.			
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:	2h PIPE_CONTROL
		Format:	OpCode
	23:16	3D Command Sub Opcode	
Default Value:		0h PIPE_CONTROL	
Format:		OpCode	
15:8	Reserved		
	Format:	MBZ	
7:0	DWord Length	Default Value:	4h DWORD_COUNT_n
		Format:	=n
	Total Length - 2. Excludes DWord (0,1).		
1	31	Reserved	
		Format:	MBZ
	30	Reserved	
		Format:	MBZ
	29	Reserved	
Format:	MBZ		
28	Reserved		
Format:	MBZ		
27	Reserved		



PIPE_CONTROL			
26	Flush LLC		
	Format:	Enable	
<p>If enabled, at the end of the current pipe-control the last level cache is cleared of all the cachelines which have been determined as being part of the Frame Buffer.</p>			
Programming Notes			
<p>SW must always program Post-Sync Operation to "Write Immediate Data" when Flush LLC is set.</p>			
25	Reserved		
	Format:	MBZ	
24	Destination Address Type		
	Defines address space of Destination Address		
	Value	Name	Description
	0h	PPGTT	Use PPGTT address space for DW write
	1h	GGTT	Use GGTT address space for DW write
	Programming Notes		
<p>Ignored if ""No Write" is selected in Operation.</p>			
23	LRI Post Sync Operation		
	Value	Name	Description
	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 specified in the Address field.
	Programming Notes		
<p>This bit causes a post sync operation with an LRI (Load Register Immediate) operation. If this bit is set then the Post-Sync Operation field must be cleared.</p>			
22	Reserved		
21	Store Data Index		
	Format:	U1	
	Description		
<p>Ring Buffer Mode Scheduling: 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 global hardware status page. This bit only applies to the Global HW status page. If this field is 1, the Destination Address Type in this command must be set to 1 (GGTT).</p> <p>Execlist Mode Scheduling: This field is valid only if the post-sync operation is not 0. If this bit is set, the store data address is index into the global hardware status page when destination</p>			



PIPE_CONTROL											
		address type in the command is set to 1 (GGTT). The store data address is index into the per-process hardware status page when destination address type in the command is set to 0 (PPGTT).									
20	Command Streamer Stall Enable										
	Format:	U1									
	If ENABLED, the sync operation will not occur until all previous flush operations pending a completion of those previous flushes will complete, including the flush produced from this command. This enables the command to act similar to the legacy MI_FLUSH command.										
19	Global Snapshot Count Reset										
	Format:	U1									
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Don't Reset</td> <td>Do not reset the snapshot counts or Statistics Counters.</td> </tr> <tr> <td>1h</td> <td>Reset</td> <td>Reset the snapshot count in Gen4 for all the units and reset the Statistics Counters except as noted above.</td> </tr> </tbody> </table>		Value	Name	Description	0h	Don't Reset	Do not reset the snapshot counts or Statistics Counters.	1h	Reset	Reset the snapshot count in Gen4 for all the units and reset the Statistics Counters except as noted above.
Value	Name	Description									
0h	Don't Reset	Do not reset the snapshot counts or Statistics Counters.									
1h	Reset	Reset the snapshot count in Gen4 for all the units and reset the Statistics Counters except as noted above.									
	Programming Notes										
	TIMESTAMP is not reset by PIPE_CONTROL with this bit set. When Post Sync Operation is set to "Write PS Depth Count" along with Global Snapshot Count Reset, PS Depth Count is Reported first before resetting the value.										
18	TLB Invalidate										
	Format:	U1									
	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 If ENABLED, PIPE_CONTROL command will flush the in flight data written out by render engine to Global Observation point on flush done. Also Requires stall bit ([20] of DW1) set.										
	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. Post Sync Operation or CS stall must be set to ensure a TLB invalidate occurs. Otherwise no cycle will occur to the TLB cache to invalidate.										
17	Reserved										
	Format:	MBZ									



PIPE_CONTROL																													
16	<p>Generic Media State Clear</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Disable</td> </tr> </table> <p>If set, all generic media state context information will be invalidated. Any state invalidated will not be saved as part of the render engine context image. The state only become valid once it is parsed by the command streamer.</p>	Format:	Disable																										
Format:	Disable																												
15:14	<p>Post Sync Operation</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th colspan="3" style="text-align: center;">Description</th> </tr> <tr> <td colspan="3">This field specifies an optional action to be taken upon completion of the synchronization operation.</td> </tr> <tr> <td colspan="3">This field must be cleared if the LRI Post-Sync Operation bit is set.</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No Write</td> <td>No write occurs as a result of this instruction. This can be used to implement a "trap" operation, etc.</td> </tr> <tr> <td>1h</td> <td>Write Immediate Data</td> <td>Write the QWord containing Immediate Data Low, High DWs to the Destination Address</td> </tr> <tr> <td>2h</td> <td>Write PS Depth Count</td> <td>Write the 64-bit PS_DEPTH_COUNT register to the Destination Address</td> </tr> <tr> <td>3h</td> <td>Write Timestamp</td> <td>Write the 64-bit TIMESTAMP register to the Destination Address</td> </tr> </tbody> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>If executed in non-secure batch buffer, the address given will be in a PPGTT address space. If in a secure ring or batch, address given will be in GGTT space</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Workaround</th> </tr> <tr> <td>Workaround : PIPECONTROL command with "Command Streamer Stall Enable" must be programmed prior to programming a PIPECONTROL command with Post Sync Op in GPGPU mode of operation (i.e when PIPELINE_SELECT command is set to GPGPU mode of operation).</td> </tr> </table>	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	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 Data	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 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. If in a secure ring or batch, address given will be in GGTT space	Workaround	Workaround : PIPECONTROL command with "Command Streamer Stall Enable" must be programmed prior to programming a PIPECONTROL command with Post Sync Op in GPGPU mode of operation (i.e when PIPELINE_SELECT command is set to GPGPU mode of 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																											
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 Data	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 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. If in a secure ring or batch, address given will be in GGTT space																													
Workaround																													
Workaround : PIPECONTROL command with "Command Streamer Stall Enable" must be programmed prior to programming a PIPECONTROL command with Post Sync Op in GPGPU mode of operation (i.e when PIPELINE_SELECT command is set to GPGPU mode of operation).																													
13	<p>Depth Stall Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit must be set when obtaining a "visible pixel" count to preclude the possible inclusion in the PS_DEPTH_COUNT value written to memory of some fraction of pixels from objects initiated after the PIPE_CONTROL command.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>3D pipeline will not stall subsequent primitives at the Depth Test stage.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>3D pipeline will stall any subsequent primitives at the Depth Test stage until the Sync and Post-Sync operations complete.</td> </tr> </tbody> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </table>	Format:	Enable	Value	Name	Description	0h	Disable	3D pipeline will not stall subsequent primitives at the Depth Test stage.	1h	Enable	3D pipeline will stall any subsequent primitives at the Depth Test stage until the Sync and Post-Sync operations complete.	Programming Notes																
Format:	Enable																												
Value	Name	Description																											
0h	Disable	3D pipeline will not stall subsequent primitives at the Depth Test stage.																											
1h	Enable	3D pipeline will stall any subsequent primitives at the Depth Test stage until the Sync and Post-Sync operations complete.																											
Programming Notes																													



PIPE_CONTROL													
		<p>This bit must be DISABLED for operations other than writing PS_DEPTH_COUNT.</p> <p>This bit will have no effect (besides preventing write cache flush) if set in a PIPE_CONTROL command issued to the Media pipe.</p>											
12	Render Target Cache Flush Enable	<table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Setting this bit will force Render Cache to be flushed to memory prior to this synchronization point completing. This bit must 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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable Flush</td> <td>Render Target Cache is NOT flushed.</td> </tr> <tr> <td>1h</td> <td>Enable Flush</td> <td>Render Target Cache is flushed.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This bit must be DISABLED for End-of-pipe (Read) fences, PS_DEPTH_COUNT or TIMESTAMP queries.</p> <p>This bit must not be set when Depth Stall Enable bit is set in this packet.</p>	Format:	Enable	Value	Name	Description	0h	Disable Flush	Render Target Cache is NOT flushed.	1h	Enable Flush	Render Target Cache is flushed.
Format:	Enable												
Value	Name	Description											
0h	Disable Flush	Render Target Cache is NOT flushed.											
1h	Enable Flush	Render Target Cache is flushed.											
11	Instruction Cache Invalidate Enable	<table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>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.</p>	Format:	Enable									
Format:	Enable												
10	Texture Cache Invalidation Enable	<table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>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.</p> <p style="text-align: center;">Workaround</p> <p>Workaround : "CS Stall" bit in PIPE_CONTROL command must be always set for GPGPU workloads when "Texture Cache Invalidation Enable" bit is set</p>	Format:	Enable									
Format:	Enable												
9	Indirect State Pointers Disable	<table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p style="text-align: center;">Description</p> <p>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.</p>	Format:	Enable									
Format:	Enable												



PIPE_CONTROL					
	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	<p>Notify Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, a Sync Completion Interrupt will be generated (if enabled by the MI Interrupt Control registers) once the sync operation is complete. See Interrupt Control Registers in Memory Interface Registers for details.</p>	Format:	Enable		
Format:	Enable				
7	<p>Pipe Control Flush Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Hardware on parsing PIPECONTROL command with Pipe Control Flush Enable set will wait for all the outstanding post sync operations corresponding to previously executed PIPECONTROL commands are complete before making forward progress.</p>	Format:	Enable		
Format:	Enable				
6	Reserved				
5	<p>DC Flush Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Setting this bit enables flushing of the L3\$ portions that caches DC writes.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>DC Flush (L3 Flush) by default doesn't result in flushing/invalidating the IA Coherent lines from L3\$, however this can be achieved by setting control bit "Pipe line flush Coherent lines" in "L3SQCREG4" register.</p>	Format:	Enable	Programming Notes	
Format:	Enable				
Programming Notes					
4	<p>VF Cache Invalidation Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Setting this bit is independent of any other bit in this packet. This bit controls the invalidation of VF address based cache at the top of the pipe i.e. at the parsing time.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>If the VF Cache Invalidation Enable is set to a 1 in a PIPE_CONTROL, a separate Null PIPE_CONTROL, all bitfields sets to 0, with the VF Cache Invalidation Enable;set to 0 needs to be sent prior to the PIPE_CONTROL with VF Cache Invalidation Enable set to a 1.</p> <p>As the VF, VFR data caches do not support a full address tag, SW shall invalidate the VF, VFR cache before a draw call that uses modified VERTEX_BUFFER state. Failure to do so may result in erroneous hits in the cache.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Workaround</td> </tr> </table>	Format:	Enable	Programming Notes	Workaround
Format:	Enable				
Programming Notes					
Workaround					



PIPE_CONTROL													
	Workaround: When VF Cache Invalidate is set "Post Sync Operation" must be enabled to "Write Immediate Data" or "Write PS Depth Count" or "Write Timestamp".												
3	Constant Cache Invalidation Enable Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>Enable</td></tr></table> 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.			Enable									
	Enable												
2	State Cache Invalidation Enable Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>Enable</td></tr></table> Setting this bit is independent of any other bit in this packet. This bit controls the invalidation of the L1 and L2 state caches at the top of the pipe i.e. at the parsing time.			Enable									
	Enable												
1	Stall At Pixel Scoreboard Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>Enable</td></tr></table> Defines the behavior of PIPE_CONTROL command at the pixel scoreboard. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Stall at the pixel scoreboard is disabled.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Stall at the pixel scoreboard is enabled.</td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p style="text-align: center; color: blue; font-weight: bold;">Programming Notes</p> <p>This bit must 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.</p> </div>			Enable	Value	Name	Description	0h	Disable	Stall at the pixel scoreboard is disabled.	1h	Enable	Stall at the pixel scoreboard is enabled.
	Enable												
Value	Name	Description											
0h	Disable	Stall at the pixel scoreboard is disabled.											
1h	Enable	Stall at the pixel scoreboard is enabled.											
0	Depth Cache Flush Enable Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>Enable</td></tr></table> Setting this bit enables flushing (i.e. writing back the dirty lines to memory and invalidating the tags) of depth related caches. This bit applies to HiZ cache, Stencil cache and depth cache. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Flush Disabled</td> <td>Depth relates caches (HiZ, Stencil and Depth) are NOT flushed.</td> </tr> <tr> <td>1h</td> <td>Flush Enabled</td> <td>Depth relates caches (HiZ, Stencil and Depth) are flushed.</td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p style="text-align: center; color: blue; font-weight: bold;">Programming Notes</p> <p>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 must be DISABLED for End-of-pipe (Read) fences, PS_DEPTH_COUNT or TIMESTAMP queries.</p> </div>			Enable	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.
	Enable												
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.											



PIPE_CONTROL				
2	31:2	<p>Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:2]U32</td> </tr> </table> <p>If Post Sync Operation is set to 1h LRI Post-Sync Operation must be clear); Bits 31:3 specify the 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.</p>	Format:	GraphicsAddress[31:2]U32
	Format:	GraphicsAddress[31:2]U32		
1:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
3	31:0	<p>Address High</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[63:32]U32</td> </tr> </table> <p>This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space. This field is valid only if the post-sync operation is not 0 and the LRI Post-Sync Operation is clear.</p>	Format:	GraphicsAddress[63:32]U32
Format:	GraphicsAddress[63:32]U32			
4..5	63:0	<p>Immediate Data</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U64</td> </tr> </table> <p>This field specifies the QWord 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. Ignored if Post-Sync Operation is "No write", "Write PS_DEPTH_COUNT" or "Write TIMESTAMP".</p>	Format:	U64
Format:	U64			



PIPELINE_SELECT

PIPELINE_SELECT			
Source:	BSpec		
Length Bias:	1		
Description			
<p>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.</p> <p>Issuing 3D-pipeline-specific commands when the GPGPU pipeline is selected, or vice versa, is UNDEFINED.</p> <p>Programming common non pipeline commands (e.g., STATE_BASE_ADDRESS) is allowed in all pipeline modes.</p>			
Programming Notes			
<p>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.</p> <p>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) PIPELINE_SELECT (GPGPU)</p>			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h GFXPIPE
		Format:	OpCode
	28:27	Command SubType	
		Default Value:	1h GFXPIPE_SINGLE_DW
		Format:	OpCode
	26:24	3D Command Opcode	
		Format:	OpCode
		Value	Name
		1h	GFXPIPE_NONPIPELINED [Default]
23:16	3D Command Sub Opcode		
	Default Value:	04h PIPELINE_SELECT	
	Format:	OpCode	



PIPELINE_SELECT		
15:8	Mask Bits	
	Programming Notes	
Must be set to modify corresponding bits in Bits 7:0. (For implemented bits)		
7	Reserved	
6	Reserved	
5	Force Media Awake	
Format:		Enable
Value	Name	Description
0	Disabled	Command streamer sends message to PM to disable force awake of media engine (next instructions do not require the media engine to be awake). Command streamer waits for acknowledge from PM before parsing the next command.
1	Enabled	Command streamer sends message to PM to force awake media engine (next instructions require media engine awake). Command streamer waits for acknowledge from PM before parsing the next command.
Programming Notes		
Mask bit [13] has to be set for HW to look at this field when PIPELINE_SELECT command is parsed. Example for usage model: RCS Ring Buffer: PIPELINE_SELECT (Force Media Awake set to '1') MI_SEMPAHORE_SINGAL (Signal context id 0xABC to Render Command Streamer) PIPELINE_SELECT (Force Media Awake set to '0') MI_BATCH_BUFFER_START STATE Commands PIPELINE_SELECT (Force Media Awake set to '1') MI_LOAD_REGISTER_IMM (Load register 0x23XX in render command streamer with data 0xFFFF) PIPELINE_SELECT (Force Media Awake set to '0') MI_BATCH_BUFFER_END		
4	Media Sampler DOP Clock Gate Enable	
Format:		Enable
Value	Name	Description
0	Disabled	Command Streamer sends message to PM to disable sampler DOP Clock Gating.
1	Enabled	Command Streamer sends message to PM to enable media sampler DOP Clock Gating.



PIPELINE_SELECT													
Programming Notes													
Mask bit [12] has to be set for HW to look at this field when PIPELINE_SELECT command is parsed.													
3	Render Sampler Power Gate Enable												
	Format:	Enable											
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled</td> <td>Command Streamer sends message to PM to disable render sampler Power Gating.</td> </tr> <tr> <td>1</td> <td>Enabled</td> <td>Command Streamer sends message to PM to enable render sampler Power Gating.</td> </tr> </tbody> </table>	Value	Name	Description	0	Disabled	Command Streamer sends message to PM to disable render sampler Power Gating.	1	Enabled	Command Streamer sends message to PM to enable render sampler Power Gating.			
Value	Name	Description											
0	Disabled	Command Streamer sends message to PM to disable render sampler Power Gating.											
1	Enabled	Command Streamer sends message to PM to enable render sampler Power Gating.											
Programming Notes													
Mask bit [11] has to be set for HW to look at this field when PIPELINE_SELECT command is parsed.													
2	Render Slice common Power Gate Enable												
	Format:	Enable											
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled</td> <td>Command Streamer sends message to PM to disable render slice common Power Gating.</td> </tr> <tr> <td>1</td> <td>Enabled</td> <td>Command Streamer sends message to PM to enable render slice common Power Gating.</td> </tr> </tbody> </table>	Value	Name	Description	0	Disabled	Command Streamer sends message to PM to disable render slice common Power Gating.	1	Enabled	Command Streamer sends message to PM to enable render slice common Power Gating.			
Value	Name	Description											
0	Disabled	Command Streamer sends message to PM to disable render slice common Power Gating.											
1	Enabled	Command Streamer sends message to PM to enable render slice common Power Gating.											
Programming Notes													
Mask bit [10] has to be set for HW to look at this field when PIPELINE_SELECT command is parsed.													
1:0	Pipeline Selection												
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>3D</td> <td>3D pipeline is selected</td> </tr> <tr> <td>1</td> <td>Media</td> <td>Media pipeline is selected (Includes HD optical disc playback, HD video playback, and generic media workloads)</td> </tr> <tr> <td>2</td> <td>GPGPU</td> <td>GPGPU pipeline is selected</td> </tr> </tbody> </table>	Value	Name	Description	0	3D	3D pipeline is selected	1	Media	Media pipeline is selected (Includes HD optical disc playback, HD video playback, and generic media workloads)	2	GPGPU	GPGPU pipeline is selected
	Value	Name	Description										
	0	3D	3D pipeline is selected										
1	Media	Media pipeline is selected (Includes HD optical disc playback, HD video playback, and generic media workloads)											
2	GPGPU	GPGPU pipeline is selected											
Programming Notes													
Mask bits [9:8] has to be set for HW to look at this field when PIPELINE_SELECT command is parsed. Setting only one of the mask bit [9] or [8] is illegal.													



Plane

pln - Plane	
Source:	Eulsa
Length Bias:	4
<p>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, q is the second component and r is the fourth component of the 4-tuple (128-bit aligned) that p belongs to.</p>	
<p>Format:</p> <p>[(pred)] pln[.cmod] (exec_size) dst src0 src1</p>	
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 : If ExecSize is 16 then accumulator cannot be used on src1.	
Syntax	
[(pred)] pln[.cmod] (exec_size) reg reg reg	
Pseudocode	
<pre> Evaluate(WrEn); for (n = 0; n < exec_size; n++) { 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 = src1.(RegNum + 1) } else { if (n < 8) { u = src1.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; } } </pre>	



pln - Plane			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	N
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	(([RegSource][Src1.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	ImmSource	
Exists If:		(([ImmSource][Src1.RegFile]='IMM')	
Format:	EU_INSTRUCTION_SOURCES_REG_IMM		
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Read Surface Info MSD

MSD_RSI - Read Surface Info MSD		
Source:	Read-Only DataPort	
Length Bias:	1	
Family:	Other	
Group:	Read Surface Info	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHF Indicates that the message forbids a header.
	18	Reserved Format: MBZ Ignored
	17:14	Message Type Default Value: 06h Format: Opcode Read Surface Info message
	13:8	Reserved Format: MBZ Ignored
	7:0	Binding Table Index Format: MDC_BTS Specifies the Binding Table Index for the message



REP16 Render Target Write MSD

MSD_RTW_REP16 - REP16 Render Target Write MSD						
Source:	Render Cache DataPort					
Length Bias:	1					
Family:	Other					
Group:	Render Target R/W					
DWord	Bit	Description				
0	31	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
	Format:	MBZ				
	30	<p>Message Precision Subtype</p> <table border="1"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Full precision data message</p>	Default Value:	0h	Format:	Opcode
	Default Value:	0h				
	Format:	Opcode				
	29	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
	Format:	MBZ				
	28:25	<p>Message Length</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.</p>	Format:	U4		
Format:	U4					
24:20	<p>Response Length</p> <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.</p>	Format:	U5			
Format:	U5					
19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the 2-register header.</p>	Format:	MDC_MHP			
Format:	MDC_MHP					
18	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					



MSD_RTW_REP16 - REP16 Render Target Write MSD					
17:14	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0Ch</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Render Target Write message</p>	Default Value:	0Ch	Format:	Opcode
Default Value:	0Ch				
Format:	Opcode				
13	<p>Per-Sample PS Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.</p> <p style="text-align: center;">Programming Notes</p> <p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE.</p> <p>When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_CO as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0.</p> <p>When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples).</p> <p>When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>	Format:	Enable		
Format:	Enable				
12	<p>Last Render Target Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.</p> <p style="text-align: center;">Programming Notes</p> <p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set. In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>	Format:	Enable		
Format:	Enable				
11	<p>Slot Group Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MDC_RT_SGS</td> </tr> </table> <p>This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.</p>	Format:	MDC_RT_SGS		
Format:	MDC_RT_SGS				



MSD_RTW_REP16 - REP16 Render Target Write MSD	
10:8	Render Target Message Subtype
	Default Value: 1h
	Format: Opcode
	SIMD16 Single source message with replicated data. Use slots [15:0] for pixel enables, X/Y addresses, and oMask.
	Programming Notes The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [31:16] are referenced instead of [15:0].
7:0	Binding Table Index
	Format: MDC_BTS Specifies the Binding Table Index for the message



Return

ret - Return			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>Return execution to the code sequence that called a subroutine.</p> <p>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.</p> <p>When SPF is on, the predication control must be scalar.</p>			
<p>Format:</p> <p>[(pred)] ret (exec_size) null src0</p>			
Restriction			
Restriction : This instruction cannot take accumulator as source.			
Syntax			
[(pred)] ret (exec_size) null reg			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { PcIP[n] = src0.chan[0]; CallMask[n] = 0; } else { PcIP[n] = IP + 1; } } for (n = exec_size; n < 32; n++) { PcIP[n] = IP + 1; } if (CallMask[n:0] == 0) { // all channels are zero Jump(src0.chan[0]); CallMask = src0.chan[1]; }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types			
D, UD			
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	[[Operand Controls][Src0.RegFile]!='IMM']
	Format:	EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource	
Exists If:		[[Operand Controls][Src0.RegFile]=='IMM']	
Format:	EU_INSTRUCTION_SOURCES_IMM32		



ret - Return		
	63:32	Operand Controls Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header Format: EU_INSTRUCTION_HEADER



Round Down

rndd - Round Down			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>Each result follows the rules in the following tables based on the floating-point mode.</p>			
Format:	[(pred)] rndd[.cmod] (exec_size) dst src0		
Syntax			
[(pred)] rndd[.cmod] (exec_size) reg reg [(pred)] rndd[.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]); } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([Operand Controls][Src0.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource	
Exists If:		([Operand Controls][Src0.RegFile]=='IMM')	
Format:	EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls		
Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Round to Nearest or Even

rnde - Round to Nearest or Even			
Source:	Eulsa		
Length Bias:	4		
<p>The <code>rnde</code> instruction takes component-wise floating point round-to-even operation of <code>src0</code> with results in two pieces - a downward rounded integral float results stored in <code>dst</code> and the round-to-even increments stored in the rounding increment bits. The round-to-even increment must be added to the results in <code>dst</code> to create the final round-to-even values to emulate the round-to-even operation, commonly known as the <code>round()</code> 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.</p> <p>Each result follows the rules in the following tables based on the floating-point mode.</p>			
<p>Format:</p> <p><code>[(pred)] rnde[.cmod] (exec_size) dst src0</code></p>			
Syntax			
<p><code>[(pred)] rnde[.cmod] (exec_size) reg reg [(pred)] rnde[.cmod] (exec_size) reg imm32</code></p>			
Pseudocode			
<pre>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	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	<code>[(Operand Controls)][Src0.RegFile]!='IMM'</code>
		Format:	EU_INSTRUCTION_SOURCES_REG
	127:64	ImmSource	
		Exists If:	<code>[(Operand Controls)][Src0.RegFile]=='IMM'</code>
		Format:	EU_INSTRUCTION_SOURCES_IMM32
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	



rnde - Round to Nearest or Even					
31:0	<table border="1"> <tr> <td colspan="2">Header</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_HEADER</td> </tr> </table>	Header		Format:	EU_INSTRUCTION_HEADER
Header					
Format:	EU_INSTRUCTION_HEADER				



Round to Zero

rndz - Round to Zero			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p>			
Format:	[[pred]] rndz[.cmod] (exec_size) dst src0		
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	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	[[Operand Controls]][Src0.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource	
Exists If:		[[Operand Controls]][Src0.RegFile]=='IMM')	
Format:	EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls		
Format:	EU_INSTRUCTION_OPERAND_CONTROLS		
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Round Up

rndu - Round Up			
Source:	Eulsa		
Length Bias:	4		
<p>The rndu instruction takes component-wise floating point upward rounding (to the integral float number closer to positive infinity) of src0, commonly known as the ceiling() function.</p> <p>Each result follows the rules in the following tables based on the floating-point mode.</p>			
<p>Format: [(pred)] rndu[.cmod] (exec_size) dst src0</p>			
Syntax			
[(pred)] rndu[.cmod] (exec_size) reg reg [(pred)] rndu[.cmod] (exec_size) reg imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { if (src0.chan[n] - floor(src0.chan[n]) > 0.0f) { dst.chan[n] = floor(src0.chan[n]) + 1; } else { dst.chan[n] = src0.chan[n]; } } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
F	F		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	!([Operand Controls][Src0.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_SOURCES_REG	
	127:64	ImmSource	
Exists If:		!([Operand Controls][Src0.RegFile]!='IMM')	
Format:	EU_INSTRUCTION_SOURCES_IMM32		
63:32	Operand Controls		
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
31:0	Header		
	Format:	EU_INSTRUCTION_HEADER	



Sampler Cache Media Block Read MSD

MSD_SC_MB - Sampler Cache Media Block Read MSD		
Source:	Read-Only DataPort	
Length Bias:	1	
Family:	Other	
Group:	Media Block R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHR Indicates that the message requires a header.
	18	Reserved
		Format: MBZ Ignored
	17:14	Message Type
		Default Value: 05h
Format: Opcode Media Block Read Sampler Cache message		
13:11	Reserved	
	Format: MBZ Ignored	
10:8	Vertical Line Stride Override	
	Format: MDC_VLSO If enabled, specifies the Vertical Line Stride and Vertical Line Stride Offset override fields.	
7:0	Binding Table Index	
	Format: MDC_BTS Specifies the Binding Table Index for the message	



Sampler Cache Oword Unaligned Block Read MSD

MSD_SC_OWUB - Sampler Cache Oword Unaligned Block Read MSD						
Source:	Read-Only DataPort					
Length Bias:	1					
Family:	Block R/W					
Group:	OW Unaligned Block R/W					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHR</td> </tr> </table> <p>Indicates that the message requires a header.</p>	Format:	MDC_MHR		
	Format:	MDC_MHR				
	18	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
	Format:	MBZ				
	17:14	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>04h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Oword Unaligned Block Read Sampler Cache message</p>	Default Value:	04h	Format:	Opcode
	Default Value:	04h				
Format:	Opcode					
13:11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ			
Format:	MBZ					
10:8	<p>Data Elements</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_DB_OW</td> </tr> </table> <p>Specifies the number of contiguous Owords to be read</p>	Format:	MDC_DB_OW			
Format:	MDC_DB_OW					
7:0	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS			
Format:	MDC_BTS					



Scaled Untyped Surface Read MSD

MSD2R_US - Scaled Untyped Surface Read MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Untyped Surface R/W	
Group:	Scaled Untyped Surface R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_A32_MHP If present, modifies the address calculations.
	18:15	Message Type
		Default Value: 01h
		Format: Opcode Untyped Surface Read message
	14	Reserved
		Format: MBZ Ignored
13:12	SIMD Mode	
	Format: MDC_SM3 Specifies the SIMD mode of the message (number of slots processed)	
11:8	Channel Mask	
	Format: MDC_CMASK Specifies which RGBA channels are included in the message payload.	
7:0	Sideband Scaled Offset	
	Format: MDC_A32_SBSO In combination with Header Present field, specifies the Scale pitch and the Offset for the message.	



Scaled Untyped Surface Write MSD

MSD2W_US - Scaled Untyped Surface Write MSD		
Source:	DataPort 2	
Length Bias:	1	
Family:	Untyped Surface R/W	
Group:	Scaled Untyped Surface R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_A32_MHP If present, modifies the address calculations.
	18:15	Message Type Default Value: 09h Format: Opcode Untyped Surface Write message
	14	Reserved Format: MBZ Ignored
	13:12	SIMD Mode Format: MDC_SM3 Specifies the SIMD mode of the message (number of slots processed)
	11:8	Channel Mask Format: MDC_UW_CMASK Specifies which RGBA channels are included in the message payload.
	7:0	Sideband Scaled Offset Format: MDC_A32_SBSO In combination with Header Present field, specifies the Scale pitch and the Offset for the message.



Scattered Move

smov - Scattered Move			
Source:	Eulsa		
Length Bias:	4		
<p>The smov instruction moves the components in src0 into dst.</p> <p>For each enabled channel, copy src0 to dst. The immediate is used to selectively enable channels without using flags.</p> <p>When predication is enabled, the predicate mask is not generated from the flags. Instead, the immediate is used to mask the execution mask. If any channel is enabled as a result of this masking, the instruction is executed.</p> <p>When predication is not enabled, the immediate masks the execution mask. This provides flexibility to mask out any channel with an immediate.</p>			
Format:	[(pred)] smov[.cmod] (exec_size) dst src0 src1		
Programming Notes			
When predication is disabled, the immediate provides the flexibility to perform a select operation without the use of flags.			
When predication is enabled, the usage model provides flexibility to select any bit in the flag registers for predication for execution size of 1.			
Syntax			
[(pred)] smov[.cmod] (exec_size) reg reg imm32			
Pseudocode			
<pre>if pred emask = OR (emask AND imm32) Else pmask = imm32. Evaluate(WrEn); for (n = 0; n < 32; n++) { if (WrEn.chan[n]) { dst.chan[n] = src0.chan[n]; } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N
Src Types		Dst Types	
*W,*D, HF, F		*W,*D, HF, F	
*W,*D, HF, F		*Q, DF	
*Q, DF		*W,*D, HF, F	
*Q, DF		*Q, DF	
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG



smov - Scattered Move		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Scratch Block Read MSD

MSDOR_SB - Scratch Block Read MSD		
Source:	DataPort 0	
Length Bias:	1	
Family:	Block R/W	
Group:	HW Block R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header.
	18	Scratch Block Message Default Value: 1h Format: Opcode Scratch Block Message
	17	Operation Type Default Value: 0h Format: Opcode Scratch Block Read message
	16	Channel Mode Format: MDC_CMODE Specifies whether the read or write operation occurs on all 4 Dwords if any of those channel enables are set, or else only on the dwords whose corresponding channel enable is set.
	15	Invalidate After Read Format: MDC_IAR Specifies if L3 cache lines accessed by the message should be invalidated after the read occurs
	14	Reserved Format: MBZ Ignored
	13:12	Data Elements Format: MDC_DB_HW Specifies the number of registers to be read or written



MSDOR_SB - Scratch Block Read MSD

	11:0	Address Offset	
		Format:	GeneralStateOffset[17:6]
		HWORD (32 byte) based address offset to the BufferAddress in the Message Header.	



Scratch Block Write MSD

MSD0W_SB - Scratch Block Write MSD		
Source:	DataPort 0	
Length Bias:	1	
Family:	Block R/W	
Group:	HW Block R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header.
	18	Scratch Block Message Default Value: 1h Format: Opcode Scratch Block Message
	17	Operation Type Default Value: 1h Format: Opcode Scratch Block Write message
	16	Channel Mode Format: MDC_CMODE Specifies whether the read or write operation occurs on all 4 Dwords if any of those channel enables are set, or else only on the dwords whose corresponding channel enable is set.
	15:14	Reserved Format: MBZ Ignored
	13:12	Data Elements Format: MDC_DB_HW Specifies the number of registers to be read or written
	11:0	Address Offset Format: GeneralStateOffset[17:6] HWORD (32 byte) based address offset to the BufferAddress in the Message Header.



Select

sel - Select	
Source:	Eulsa
Length Bias:	4
Description	
<p>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.</p> <p>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.</p> <p>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.</p> <p>A sel instruction with cmod .l is used to emulate a MIN instruction.</p> <p>A sel instruction with cmod .ge is used to emulate a MAX instruction.</p> <p>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.</p> <p>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). This applies even if the source modifiers are set on the sel instruction sources.</p> <p>The sel instruction uses any conditional modifier internally and does not update the flag register if a conditional modifier is used.</p> <p>A sel instruction with cmod or source modifier will flush denorm to zero, depending on the denorm mode bit; a sel instruction without cmod and source modifier will retain denorm.</p>	
Format: (pred) sel[.cmod] (exec_size) dst src0 src1	
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>	
Predication	Conditional Modifier
Saturation	Source Modifier



sel - Select			
Y	Y	Y	Y
Src Types		Dst Types	
*B,*W,*D		*B,*W,*D	
F		F	
DF		DF	
*W,*D		*W,*D	
*W,*D		*Q	
*Q		*W,*D	
*Q		*Q	
HF		HF	
HF, F		HF, F	
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	(([RegSource][Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG
	127:64	ImmSource	
		Exists If:	(([ImmSource][Src1.RegFile]='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_IMM
	63:32	Operand Controls	
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header	
		Format:	EU_INSTRUCTION_HEADER



Send Message

send - Send Message	
Source:	Eulsa
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>.	
<p>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) and the Response Length (the number of consecutive GRF registers). It also contains the header present bit, and the function control signals. The extend message descriptor field <ex_desc> contains the target function ID. WrEn is forwarded to the target function in the message sideband.</p>	
The extended message descriptor field <ex_desc> also contains the extended function control field to be sent to the Target Shared Function over message sideband.	
<p>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.</p>	
<p><ex_desc> is a 32-bit immediate, imm32. The lower 4bits of the <ex_desc> specifies the SFID for the message. The bit5 of the extended message descriptor, the EOT field, always comes from bit 127 of the instruction word. A thread must terminate with a send instruction with EOT turned on. The higher 16bits, bit31:16 specify the 16bit extended function control field. Interpretation of the extended function control signals is subject to the target external function.</p>	
<src> is a 256-bit aligned GRF register. It serves as the leading GRF register of the request.	
The source dependency control, {NoSrcDepSet} is used to control the setting of source dependency for the source.	
<p><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</p>	



send - Send Message

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 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. 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 encodings Switch is 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 thread into the same slot while the message with EOT for current thread is pending dispatch. Any instruction updating the ARF must use a {Switch} if the ARF is not used before EOT. DepCtrl Must not be used with Send Instruction. When pagefault is enabled, the source and destination operands must not overlap. This is required to ensure the messages can be replayed.</src>

Restriction :

The source dependency control, {NoSrcDepSet}, must not be set for the send instruction preceding a send instruction with EOT.

Syntax

```
[(pred)] send (exec_size) reg reg imm32 reg32a [(pred)] send (exec_size) reg reg
imm32 imm32
```

Pseudocode

```
Evaluate(WrEn); <MsgChEnable> = WrEn; <SourceReg> = <src>.RegNum;
MessageEnqueue(<MsgChEnable>, <ResponseReg>, <SourceReg>, <desc>, <ex_dest>);
```

Predication	Conditional Modifier	Saturation	Source Modifier
--------------------	-----------------------------	-------------------	------------------------



send - Send Message			
Y	N	N	N
DWord	Bit	Description	
0..3	127:96	Message	
		Format:	EU_INSTRUCTION_OPERAND_SEND_MSG
	95:89	Flags	
		Format:	EU_INSTRUCTION_FLAGS
	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[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16')
		Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
	63:32	Operand Control	
	Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
31:28	Controls B		
	Format:	EU_INSTRUCTION_CONTROLS_B	
27:24	Shared Function ID (SFID)		
	Format:	SFID	
23:8	Controls A		
	Format:	EU_INSTRUCTION_CONTROLS_A	
7	Reserved		
	Format:	MBZ	
6:0	Opcode		
	Format:	EU_OPCODE	



SFC_AVS_CHROMA_Coeff_Table

SFC_AVS_CHROMA_Coeff_Table			
Source:	BSpec		
Length Bias:	2		
This command is sent from VDBOX/VEBOX to SFC pipeline at the start of each frame once the lock request is granted.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value: 3h PARALLEL_VIDEO_PIPE	
		Format: OpCode	
	28:27	Pipeline	
		Default Value: 2h Media	
		Format: OpCode	
	26:23	Media Command Opcode	
		Format: OpCode	
		Value	Name
		Ah	Media Misc [Default]
22:21	SubOpcodeA		
	Default Value: 0h Common		
	Format: OpCode		
20:16	SubOpcodeB		
	Default Value: 5h SFC_AVS CHROMA Coeff_Table		
	Format: OpCode		
15:12	Reserved		
	Format: MBZ		
11:0	DWord Length		
	Default Value: 7Fh Excludes DWord (0,1)		
	Format: =n		
	Total Length - 2		
1..2	63:56	Table 1Y Filter Coefficient[[n],5]	
		Format: S1.6 2's Complement	
		Range: [-2, +2)	
		Chroma table for Y-direction.	



SFC_AVS_CHROMA_Coeff_Table													
	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.									
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
55:48	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 1X Filter Coefficient[[n],5]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2]</td> </tr> <tr> <td colspan="2">Chroma table for X-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 1X Filter Coefficient[[n],5]		Format:	S1.6 2's Complement	Range: [-2, +2]		Chroma table for X-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 1X Filter Coefficient[[n],5]													
Format:	S1.6 2's Complement												
Range: [-2, +2]													
Chroma table for X-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
47:40	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 1Y Filter Coefficient[[n],4]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2]</td> </tr> <tr> <td colspan="2">Chroma table for Y-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 1Y Filter Coefficient[[n],4]		Format:	S1.6 2's Complement	Range: [-2, +2]		Chroma table for Y-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 1Y Filter Coefficient[[n],4]													
Format:	S1.6 2's Complement												
Range: [-2, +2]													
Chroma table for Y-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
39:32	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 1X Filter Coefficient[[n],4]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2]</td> </tr> <tr> <td colspan="2">Chroma table for X-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 1X Filter Coefficient[[n],4]		Format:	S1.6 2's Complement	Range: [-2, +2]		Chroma table for X-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 1X Filter Coefficient[[n],4]													
Format:	S1.6 2's Complement												
Range: [-2, +2]													
Chroma table for X-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
31:24	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 1Y Filter Coefficient[[n],3]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2]</td> </tr> <tr> <td colspan="2">Chroma table for Y-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 1Y Filter Coefficient[[n],3]		Format:	S1.6 2's Complement	Range: [-2, +2]		Chroma table for Y-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 1Y Filter Coefficient[[n],3]													
Format:	S1.6 2's Complement												
Range: [-2, +2]													
Chroma table for Y-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
23:16	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 1X Filter Coefficient[[n],3]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table>	Table 1X Filter Coefficient[[n],3]		Format:	S1.6 2's Complement								
Table 1X Filter Coefficient[[n],3]													
Format:	S1.6 2's Complement												



SFC_AVS_CHROMA_Coeff_Table				
		<p>Range: [-2, +2] Chroma table for X-direction.</p> <p style="text-align: center;">Programming Notes</p> <p>For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</p>		
	15:8	<p>Table 1Y Filter Coefficient[[n],2]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> <p>Range: [-2, +2] Chroma table for Y-direction.</p> <p style="text-align: center;">Programming Notes</p> <p>For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</p>	Format:	S1.6 2's Complement
Format:	S1.6 2's Complement			
	7:0	<p>Table 1X Filter Coefficient[[n],2]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> <p>Range: [-2, +2] Chroma table for X-direction.</p> <p style="text-align: center;">Programming Notes</p> <p>For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</p>	Format:	S1.6 2's Complement
Format:	S1.6 2's Complement			
3.64	1983:0	<p>Filter Coefficients</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>Chroma_Filter_Coefficients_Array[31]</td> </tr> </table>	Format:	Chroma_Filter_Coefficients_Array[31]
Format:	Chroma_Filter_Coefficients_Array[31]			



SFC_AVS_LUMA_Coeff_Table

SFC_AVS_LUMA_Coeff_Table			
Source:	BSpec		
Length Bias:	2		
This command is sent from VDBOX/VEBOX to SFC pipeline at the start of each frame once the lock request is granted.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value: 3h PARALLEL_VIDEO_PIPE	
		Format: OpCode	
	28:27	Pipeline	
		Default Value: 2h Media	
		Format: OpCode	
	26:23	Media Command Opcode	
		Format: OpCode	
		Value	Name
		Ah	Media Misc [Default]
22:21	SubOpcodeA		
	Default Value: 0h Common		
	Format: OpCode		
20:16	SubOpcodeB		
	Default Value: 6h SFC_AVS LUMA Coeff_Table		
	Format: OpCode		
15:12	Reserved		
	Format: MBZ		
11:0	DWord Length		
	Format: =n		
	Total Length - 2		
	Value	Name	
	7Fh	Excludes DWord (0,1) [Default]	
1..4	127:120	Table 0Y Filter Coefficient[[n],7]	
		Format: S1.6 2's Complement	
		Range: [-2, +2)	
		Luma table for Y-direction.	



SFC_AVS_LUMA_Coeff_Table													
	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.									
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
119:112	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 0X Filter Coefficient[[n],7]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for X-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 0X Filter Coefficient[[n],7]		Format:	S1.6 2's Complement	Range: [-2, +2)		Luma table for X-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 0X Filter Coefficient[[n],7]													
Format:	S1.6 2's Complement												
Range: [-2, +2)													
Luma table for X-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
111:104	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 0Y Filter Coefficient[[n],6]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for Y-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 0Y Filter Coefficient[[n],6]		Format:	S1.6 2's Complement	Range: [-2, +2)		Luma table for Y-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 0Y Filter Coefficient[[n],6]													
Format:	S1.6 2's Complement												
Range: [-2, +2)													
Luma table for Y-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
103:96	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 0X Filter Coefficient[[n],6]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for X-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 0X Filter Coefficient[[n],6]		Format:	S1.6 2's Complement	Range: [-2, +2)		Luma table for X-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 0X Filter Coefficient[[n],6]													
Format:	S1.6 2's Complement												
Range: [-2, +2)													
Luma table for X-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
95:88	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 0Y Filter Coefficient[[n],5]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for Y-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 0Y Filter Coefficient[[n],5]		Format:	S1.6 2's Complement	Range: [-2, +2)		Luma table for Y-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 0Y Filter Coefficient[[n],5]													
Format:	S1.6 2's Complement												
Range: [-2, +2)													
Luma table for Y-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
87:80	<table border="1" style="width: 100%;"> <tr> <th colspan="2">Table 0X Filter Coefficient[[n],5]</th> </tr> <tr> <td style="width: 30%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table>	Table 0X Filter Coefficient[[n],5]		Format:	S1.6 2's Complement								
Table 0X Filter Coefficient[[n],5]													
Format:	S1.6 2's Complement												



SFC_AVS_LUMA_Coeff_Table	
	<p>Range: [-2, +2)</p> <p>Luma table for X-direction.</p> <p style="text-align: center;">Programming Notes</p> <p>For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</p>
79:72	<p>Table 0Y Filter Coefficient[[n],4]</p> <p>Format: S1.6 2's Complement</p> <p>Range: [-2, +2)</p> <p>Luma table for Y-direction.</p> <p style="text-align: center;">Programming Notes</p> <p>For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</p>
71:64	<p>Table 0X Filter Coefficient[[n],4]</p> <p>Format: S1.6 2's Complement</p> <p>Range: [-2, +2)</p> <p>Luma table for X-direction.</p> <p style="text-align: center;">Programming Notes</p> <p>For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</p>
63:56	<p>Table 0Y Filter Coefficient[[n],3]</p> <p>Format: S1.6 2's Complement</p> <p>Range: [-2, +2)</p> <p>Luma table for Y-direction.</p> <p style="text-align: center;">Programming Notes</p> <p>For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</p>
55:48	<p>Table 0X Filter Coefficient[[n],3]</p> <p>Format: S1.6 2's Complement</p> <p>Range: [-2, +2)</p> <p>Luma table for X-direction.</p> <p style="text-align: center;">Programming Notes</p> <p>For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</p>
47:40	<p>Table 0Y Filter Coefficient[[n],2]</p>



SFC_AVS_LUMA_Coeff_Table													
	<table border="1"> <tr> <td>Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for Y-direction.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Format:	S1.6 2's Complement	Range: [-2, +2)		Luma table for Y-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.			
Format:	S1.6 2's Complement												
Range: [-2, +2)													
Luma table for Y-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
39:32	<table border="1"> <tr> <td colspan="2">Table 0X Filter Coefficient[[n],2]</td> </tr> <tr> <td>Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for X-direction.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 0X Filter Coefficient[[n],2]		Format:	S1.6 2's Complement	Range: [-2, +2)		Luma table for X-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 0X Filter Coefficient[[n],2]													
Format:	S1.6 2's Complement												
Range: [-2, +2)													
Luma table for X-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
31:24	<table border="1"> <tr> <td colspan="2">Table 0Y Filter Coefficient[[n],1]</td> </tr> <tr> <td>Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for Y-direction.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 0Y Filter Coefficient[[n],1]		Format:	S1.6 2's Complement	Range: [-2, +2)		Luma table for Y-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 0Y Filter Coefficient[[n],1]													
Format:	S1.6 2's Complement												
Range: [-2, +2)													
Luma table for Y-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
23:16	<table border="1"> <tr> <td colspan="2">Table 0X Filter Coefficient[[n],1]</td> </tr> <tr> <td>Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for X-direction.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Table 0X Filter Coefficient[[n],1]		Format:	S1.6 2's Complement	Range: [-2, +2)		Luma table for X-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
Table 0X Filter Coefficient[[n],1]													
Format:	S1.6 2's Complement												
Range: [-2, +2)													
Luma table for X-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
15:8	<table border="1"> <tr> <td colspan="2">Table 0Y Filter Coefficient[[n],0]</td> </tr> <tr> <td>Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range: [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for Y-direction.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table>	Table 0Y Filter Coefficient[[n],0]		Format:	S1.6 2's Complement	Range: [-2, +2)		Luma table for Y-direction.		Programming Notes			
Table 0Y Filter Coefficient[[n],0]													
Format:	S1.6 2's Complement												
Range: [-2, +2)													
Luma table for Y-direction.													
Programming Notes													



SFC_AVS_LUMA_Coeff_Table		
		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.
	7:0	Table 0X Filter Coefficient[[n],0] Format: S1.6 2's Complement Range: [-2, +2) Luma table for X-direction. <div style="text-align: center;">Programming Notes</div> For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.
5..128	3967:0	Filter Coefficients Format: Luma_Filter_Coefficients_Array[31]



SFC_AVS_STATE

SFC_AVS_STATE							
Source:	BSpec						
Length Bias:	2						
This command is sent from VDBOX/VEBOX to SFC pipeline at the start of each frame once the lock request is granted.							
DWord	Bit	Description					
0	31:29	Command Type					
		Default Value:	3h PARALLEL_VIDEO_PIPE				
		Format:	OpCode				
	28:27	Pipeline					
		Default Value:	2h Media				
		Format:	OpCode				
	26:23	Media Command Opcode					
		Format:	OpCode				
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>Ah</td> <td>Media Misc [Default]</td> </tr> </tbody> </table>		Value	Name	Ah	Media Misc [Default]
		Value	Name				
Ah	Media Misc [Default]						
22:21	SubOpcodeA						
	Default Value:	0h Common					
	Format:	OpCode					
20:16	SubOpcodeB						
	Default Value:	2h SFC_AVS_STATE					
	Format:	OpCode					
15:12	Reserved						
	Format:	MBZ					
11:0	DWord Length						
	Default Value:	1h Excludes DWord (0,1)					
	Format:	=n					
	Total Length - 2						
1	31:24	Sharpness Level					
		Format:	U8				
		When adaptive scaling is off, determines the balance between sharp and smooth scalars.					
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Contribute 1 from the smooth scalar</td> </tr> </tbody> </table>		Value	Name	Description	0
Value	Name	Description					
0		Contribute 1 from the smooth scalar					



SFC_AVS_STATE		
	255	Contribute 1 from the sharp scalar
	23:7	Reserved Format: MBZ
	6:4	Transition Area with 4 Pixels Format: U3 Used in adaptive filtering to specify the width of the transition area for the 4 pixel calculation.
	3	Reserved Format: MBZ
	2:0	Transition Area with 8 Pixels Format: U3 Used in adaptive filtering to specify the width of the transition area for the 8 pixel calculation.
	2	31:24
23:16		Max Derivative 4 Pixels Format: U8 Used in adaptive filtering to specify the lower boundary of the smooth 4 pixel area.
15:8		Reserved Format: MBZ
7:0		MAX Derivative Point 8 Format: U8 Used in adaptive filtering to specify the lower boundary of the smooth 8 pixel area.



SFC_FRAME_START

SFC_FRAME_START			
Source:	BSpec		
Length Bias:	2		
This command is sent from VDBOX/VEBOX to SFC pipeline at the start of each frame once the lock request is granted.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value: 3h PARALLEL_VIDEO_PIPE	
		Format: OpCode	
	28:27	Pipeline	
		Default Value: 2h Media	
		Format: OpCode	
	26:23	Media Command Opcode	
		Format: OpCode	
		Value	Name
		Ah	Media Misc [Default]
22:21	SubOpcodeA		
	Default Value: 0h Common		
	Format: OpCode		
20:16	SubOpcodeB		
	Default Value: 4h SFC_FRAME_START		
	Format: OpCode		
15:12	Reserved		
	Format: MBZ		
11:0	DWord Length		
	Default Value: 0h Excludes DWord (0,1)		
	Format: =n		
	Total Length - 2		
1	31:0	Reserved	
		Format: MBZ	



SFC_IEF_STATE

SFC_IEF_STATE			
Source:	BSpec		
Length Bias:	2		
This command is sent from VDBOX/VEBOX to SFC pipeline at the start of each frame once the lock request is granted.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:23	Media Command Opcode	
		Format:	OpCode
		Value	Name
		Ah	Media Misc [Default]
22:21	SubOpcodeA		
	Default Value:	0h Common	
	Format:	OpCode	
20:16	SubOpcodeB		
	Default Value:	3h SFC_IEF_STATE	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	16h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31:28	Reserved	
		Format:	MBZ
	27:23	R3c Coefficient	
Default Value:		5	
Format:		U0.5	
IEF smoothing coefficient, <i>see IEF map.</i>			



SFC_IEF_STATE						
	22:18	<p>R3x Coefficient</p> <table border="1"> <tr> <td>Default Value:</td> <td>5</td> </tr> <tr> <td>Format:</td> <td>U0.5</td> </tr> </table> <p>IEF smoothing coefficient, <i>see IEF map</i>.</p>	Default Value:	5	Format:	U0.5
	Default Value:	5				
	Format:	U0.5				
	17:12	<p>Strong Edge Threshold</p> <table border="1"> <tr> <td>Default Value:</td> <td>8</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>If EM > Strong Edge Threshold → the basic VSA detects a strong edge.</p>	Default Value:	8	Format:	U6
Default Value:	8					
Format:	U6					
11:6	<p>Weak Edge Threshold</p> <table border="1"> <tr> <td>Default Value:</td> <td>1</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>If Strong Edge Threshold > EM > Weak Edge Threshold → the basic VSA detects a weak edge.</p>	Default Value:	1	Format:	U6	
Default Value:	1					
Format:	U6					
5:0	<p>Gain Factor</p> <table border="1"> <tr> <td>Default Value:</td> <td>44</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>User control sharpening strength.</p>	Default Value:	44	Format:	U6	
Default Value:	44					
Format:	U6					
2	31:27	<p>R5c Coefficient</p> <table border="1"> <tr> <td>Default Value:</td> <td>7</td> </tr> <tr> <td>Format:</td> <td>U0.5</td> </tr> </table> <p>IEF smoothing coefficient, <i>see IEF map</i>.</p>	Default Value:	7	Format:	U0.5
	Default Value:	7				
	Format:	U0.5				
	26:22	<p>R5cx Coefficient</p> <table border="1"> <tr> <td>Default Value:</td> <td>7</td> </tr> <tr> <td>Format:</td> <td>U0.5</td> </tr> </table> <p>IEF smoothing coefficient, <i>see IEF map</i>.</p>	Default Value:	7	Format:	U0.5
Default Value:	7					
Format:	U0.5					
21:17	<p>R5x Coefficient</p> <table border="1"> <tr> <td>Default Value:</td> <td>7</td> </tr> <tr> <td>Format:</td> <td>U0.5</td> </tr> </table> <p>IEF smoothing coefficient, <i>see IEF map</i>.</p>	Default Value:	7	Format:	U0.5	
Default Value:	7					
Format:	U0.5					
16:14	<p>Strong Edge Weight</p> <table border="1"> <tr> <td>Default Value:</td> <td>7</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Sharpening strength when a <u>STRONG</u> edge is found in basic VSA.</p>	Default Value:	7	Format:	U3	
Default Value:	7					
Format:	U3					



SFC_IEF_STATE						
3	13:11	<p>Regular Weight</p> <table border="1"> <tr> <td>Default Value:</td> <td>2</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Sharpening strength when a <u>WEAK</u> edge is found in basic VSA.</p>	Default Value:	2	Format:	U3
	Default Value:	2				
	Format:	U3				
	10:8	<p>Non Edge Weight</p> <table border="1"> <tr> <td>Default Value:</td> <td>1</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>. Sharpening strength when <u>NO EDGE</u> is found in basic VSA.</p>	Default Value:	1	Format:	U3
	Default Value:	1				
	Format:	U3				
	7:0	<p>Global Noise Estimation</p> <table border="1"> <tr> <td>Default Value:</td> <td>255</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Global noise estimation of previous frame.</p>	Default Value:	255	Format:	U8
	Default Value:	255				
	Format:	U8				
31:28	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
27:22	<p>Hue_Max</p> <table border="1"> <tr> <td>Default Value:</td> <td>14</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Rectangle half width.</p>	Default Value:	14	Format:	U6	
Default Value:	14					
Format:	U6					
21:16	<p>Sat_Max</p> <table border="1"> <tr> <td>Default Value:</td> <td>31</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Rectangle half length.</p>	Default Value:	31	Format:	U6	
Default Value:	31					
Format:	U6					
15:8	<table border="1"> <tr> <td>Format:</td> <td>S0.7 2's Complement</td> </tr> </table> <p>Default Value = 79/128</p>	Format:	S0.7 2's Complement			
Format:	S0.7 2's Complement					
7:0	<table border="1"> <tr> <td>Format:</td> <td>S0.7 2's Complement</td> </tr> </table> <p>Default Value = 101/128</p>	Format:	S0.7 2's Complement			
Format:	S0.7 2's Complement					
4	31:24	<p>V_Mid</p> <table border="1"> <tr> <td>Default Value:</td> <td>154</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Rectangle middle-point V coordinate.</p>	Default Value:	154	Format:	U8
Default Value:	154					
Format:	U8					



SFC_IEF_STATE										
	23:16	U_Mid <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>110</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Rectangle middle-point U coordinate.	Default Value:	110	Format:	U8				
	Default Value:	110								
	Format:	U8								
	15	VY_STD_Enable <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> Enables STD in the VY subspace.	Format:	Enable						
	Format:	Enable								
14:12	Diamond Margin <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>4</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table>	Default Value:	4	Format:	U3					
Default Value:	4									
Format:	U3									
11	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
10:0	S3U <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S2.8 -2's Complement</td> </tr> </table> Slope 3 of the upper part of the detection PWLF. Default: 0/256	Format:	S2.8 -2's Complement							
Format:	S2.8 -2's Complement									
5	31	Skin Detail Factor <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U1 Enumerated Type</td> </tr> </table> This flag bit is in operation only when one of the following conditions exists: <ul style="list-style-type: none"> when the control bit SkinToneTunedIEF_Enable is on. When SkinDetailFactor is equal to 0, sign(SkinDetailFactor) is equal to +1, and the content of the detected skin tone area is detail revealed. When SkinDetailFactor is equal to 1, sign(SkinDetailFactor) is equal to -1, and the content of the detected skin tone area is not detail revealed. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Detail Revealed [Default]</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Not Detail Revealed</td> </tr> </tbody> </table>	Format:	U1 Enumerated Type	Value	Name	0	Detail Revealed [Default]	1	Not Detail Revealed
		Format:	U1 Enumerated Type							
		Value	Name							
0	Detail Revealed [Default]									
1	Not Detail Revealed									
30:24	Diamond_du <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S6 -2's Complement</td> </tr> </table> Rhombus center shift in the sat-direction, relative to the rectangle center.	Default Value:	0	Format:	S6 -2's Complement					
Default Value:	0									
Format:	S6 -2's Complement									



SFC_IEF_STATE					
	<p>23:21 HS_margin</p> <table border="1"> <tr> <td>Default Value:</td> <td>3</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Defines rectangle margin.</p>	Default Value:	3	Format:	U3
	Default Value:	3			
	Format:	U3			
	<p>20:13 Diamond_alpha</p> <table border="1"> <tr> <td>Format:</td> <td>U2.6</td> </tr> </table> <p>$1 / \tan(\beta)$</p> <p>Default: 100/64</p>	Format:	U2.6		
Format:	U2.6				
<p>12:7 Diamond_Th</p> <table border="1"> <tr> <td>Default Value:</td> <td>35</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Half length of the rhombus axis in the sat-direction.</p>	Default Value:	35	Format:	U6	
Default Value:	35				
Format:	U6				
<p>6:0 Diamond_dv</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S6 -2's Complement</td> </tr> </table> <p>Rhombus center shift in the hue-direction, relative to the rectangle center.</p>	Default Value:	0	Format:	S6 -2's Complement	
Default Value:	0				
Format:	S6 -2's Complement				
6	<p>31:24 Y_point_4</p> <table border="1"> <tr> <td>Default Value:</td> <td>255</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Fourth point of the Y piecewise linear membership function.</p>	Default Value:	255	Format:	U8
	Default Value:	255			
	Format:	U8			
<p>23:16 Y_point_3</p> <table border="1"> <tr> <td>Default Value:</td> <td>254</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Third point of the Y piecewise linear membership function.</p>	Default Value:	254	Format:	U8	
Default Value:	254				
Format:	U8				
<p>15:8 Y_point_2</p> <table border="1"> <tr> <td>Default Value:</td> <td>47</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Second point of the Y piecewise linear membership function.</p>	Default Value:	47	Format:	U8	
Default Value:	47				
Format:	U8				



SFC_IEF_STATE		
	7:0	Y_point_1 Default Value: 46 Format: U8 First point of the Y piecewise linear membership function.
		Reserved Format: MBZ
7	31:16	Reserved Format: MBZ
	15:0	INV_Margin_VYL Format: U0.16 1 / Margin_VYL Default: 3300/65536
8	31:24	P1L Default Value: 216 Format: U8 Y Point 1 of the lower part of the detection PWLF.
	23:16	P0L Default Value: 46 Format: U8 Y Point 0 of the lower part of the detection PWLF.
	15:0	INV_Margin_VYU Format: U0.16 1 / Margin_VYL Default: 1600/65536
9	31:24	B1L Default Value: 130 Format: U8 V Bias 1 of the lower part of the detection PWLF.
	23:16	B0L Default Value: 133 Format: U8 V Bias 0 of the lower part of the detection PWLF.



SFC_IEF_STATE						
	15:8	<p>P3L</p> <table border="1"> <tr> <td>Default Value:</td> <td>236</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Y Point 3 of the lower part of the detection PWLF.</p>	Default Value:	236	Format:	U8
	Default Value:	236				
Format:	U8					
	7:0	<p>P2L</p> <table border="1"> <tr> <td>Default Value:</td> <td>236</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Y Point 2 of the lower part of the detection PWLF.</p>	Default Value:	236	Format:	U8
Default Value:	236					
Format:	U8					
10	31:27	<p>Y_Slope_2</p> <table border="1"> <tr> <td>Format:</td> <td>U2.3</td> </tr> </table> <p>Slope between points Y3 and Y4.</p> <p>Default: 31/8</p>	Format:	U2.3		
	Format:	U2.3				
	26:16	<p>S0L</p> <table border="1"> <tr> <td>Format:</td> <td>S2.8 -2's Complement</td> </tr> </table> <p>Slope 0 of the lower part of the detection PWLF.</p> <p>Default: -5/256</p>	Format:	S2.8 -2's Complement		
	Format:	S2.8 -2's Complement				
15:8	<p>B3L</p> <table border="1"> <tr> <td>Default Value:</td> <td>130</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 3 of the lower part of the detection PWLF.</p>	Default Value:	130	Format:	U8	
Default Value:	130					
Format:	U8					
7:0	<p>B2L</p> <table border="1"> <tr> <td>Default Value:</td> <td>130</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 2 of the lower part of the detection PWLF.</p>	Default Value:	130	Format:	U8	
Default Value:	130					
Format:	U8					
11	31:22	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
21:11	<p>S2L</p> <table border="1"> <tr> <td>Format:</td> <td>S2.8 -2's Complement</td> </tr> </table> <p>Default: 0/256</p> <p>Slope 2 of the lower part of the detection PWLF.</p>	Format:	S2.8 -2's Complement			
Format:	S2.8 -2's Complement					



SFC_IEF_STATE		
	10:0	S1L Format: S2.8 -2's Complement Default: 0/256 Slope 1 of the lower part of the detection PWLF.
		Y_Slope1 Format: U2.3 Slope between points Y1 and Y2. Default: 31/8
12	26:19	P1U Default Value: 66 Format: U8 Y Point 1 of the upper part of the detection PWLF.
		P0U Default Value: 46 Format: U8 Y Point 0 of the upper part of the detection PWLF.
	10:0	S3L Format: S2.8 -2's Complement Slope 3 of the lower part of the detection PWLF. Default: 0/256
		B1U Default Value: 163 Format: U8 V Bias 1 of the upper part of the detection PWLF.
13	23:16	B0U Default Value: 143 Format: U8 V Bias 0 of the upper part of the detection PWLF.



SFC_IEF_STATE						
	15:8	P3U <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>236</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Y Point 3 of the upper part of the detection PWLF.	Default Value:	236	Format:	U8
	Default Value:	236				
Format:	U8					
	7:0	P2U <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>150</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Y Point 2 of the upper part of the detection PWLF.	Default Value:	150	Format:	U8
Default Value:	150					
Format:	U8					
14	31:27	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	26:16	S0U <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S2.8 -2's Complement</td> </tr> </table> Slope 0 of the upper part of the detection PWLF. Default: 256/256	Format:	S2.8 -2's Complement		
	Format:	S2.8 -2's Complement				
15:8	B3U <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>140</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> V Bias 3 of the upper part of the detection PWLF.	Default Value:	140	Format:	U8	
Default Value:	140					
Format:	U8					
7:0	B2U <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>200</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> V Bias 2 of the upper part of the detection PWLF.	Default Value:	200	Format:	U8	
Default Value:	200					
Format:	U8					
15	31:22	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
21:11	S2U <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S2.8 -2's Complement</td> </tr> </table> Slope 2 of the upper part of the detection PWLF. Default: -179/256	Format:	S2.8 -2's Complement			
Format:	S2.8 -2's Complement					



SFC_IEF_STATE		
	10:0	S1U Format: S2.8 -2's Complement Default: 113/256 Slope 1 of the upper part of the detection PWLF.
		Reserved Format: MBZ
16	28:16	C1 Default Value: 0 Format: S2.10 -2's Complement Transform coefficient
		C0 Default Value: 1024 Format: S2.10 -2's Complement Transform coefficient
		Reserved Format: MBZ
	15:3	YUV Channel Swap
	2	Transform Enable
	1	Reserved
17	31:26	Reserved Format: MBZ
		C3 Default Value: 0 Format: S2.10 -2's Complement Transform coefficient
		C2 Default Value: 0 Format: S2.10 -2's Complement Transform coefficient
18	31:26	Reserved Format: MBZ



SFC_IEF_STATE						
	25:13	C5 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%;">0</td> </tr> <tr> <td>Format:</td> <td>S2.10 -2's Complement</td> </tr> </table> Transform coefficient	Default Value:	0	Format:	S2.10 -2's Complement
	Default Value:	0				
Format:	S2.10 -2's Complement					
	12:0	C4 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%;">1024</td> </tr> <tr> <td>Format:</td> <td>S2.10 -2's Complement</td> </tr> </table> Transform coefficient	Default Value:	1024	Format:	S2.10 -2's Complement
Default Value:	1024					
Format:	S2.10 -2's Complement					
19	31:26	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	25:13	C7 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%;">0</td> </tr> <tr> <td>Format:</td> <td>S2.10 -2's Complement</td> </tr> </table> Transform coefficient	Default Value:	0	Format:	S2.10 -2's Complement
Default Value:	0					
Format:	S2.10 -2's Complement					
12:0	C6 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%;">0</td> </tr> <tr> <td>Format:</td> <td>S2.10 -2's Complement</td> </tr> </table> Transform coefficient	Default Value:	0	Format:	S2.10 -2's Complement	
Default Value:	0					
Format:	S2.10 -2's Complement					
20	31:13	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
12:0	C8 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%;">1024</td> </tr> <tr> <td>Format:</td> <td>S2.10 -2's Complement</td> </tr> </table> Transform coefficient	Default Value:	1024	Format:	S2.10 -2's Complement	
Default Value:	1024					
Format:	S2.10 -2's Complement					
21	31:22	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
21:11	Offset out 1 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%;">0</td> </tr> <tr> <td>Format:</td> <td>S2.8 -2's Complement</td> </tr> </table> Offset out for Y/R.	Default Value:	0	Format:	S2.8 -2's Complement	
Default Value:	0					
Format:	S2.8 -2's Complement					



SFC_IEF_STATE			
	10:0	Offset in 1	
		Default Value: 0	
		Format: S2.8 -2's Complement	
Offset in for Y/R.			
22	31:22	Reserved	
		Format: MBZ	
	21:11	Offset out 2	
		Default Value: 0	
		Format: S2.8 -2's Complement	
	Offset out for U/G.		
	10:0	Offset in 2	
		Default Value: 0	
		Format: S2.8 -2's Complement	
Offset in for U/G.			
23	31:22	Reserved	
		Format: MBZ	
	21:11	Offset out 3	
		Default Value: 0	
		Format: S2.8 -2's Complement	
	Offset out for V/B.		
	10:0	Offset in 3	
		Default Value: 0	
		Format: S2.8 -2's Complement	
Offset in for V/B.			



SFC_LOCK

SFC_LOCK			
Source:		BSpec	
Length Bias:		2	
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:23	Media Command Opcode	
		Format:	OpCode
		Value	Name
		Ah	Media Misc [Default]
22:21	SubOpcodeA		
	Default Value:	0h Common	
	Format:	OpCode	
20:16	SubOpcodeB		
	Default Value:	0h SFC Lock	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Default Value:	0h Excludes DWord (0,1)	
	Format:	=n	
	Total Length - 2		
1	31:2	Reserved	
		Format:	MBZ
1	1	Pre-Scaled Output Surface Output Enable VD - Reconstructed Pixel Output Enable For VD Mode, this field specifies the enabling of writing out the display reconstructed pixel to memory. It could be pre or post- ILDB filter pixel output based on the pre- and post- filter setting in the AVC state command.	



SFC_LOCK																							
		<table border="1"> <thead> <tr> <th>Pre-Deblock Flag</th> <th>Post-Deblock Flag</th> <th>VD Pixels Output to Memory</th> <th>VD Pixels Output to SFC</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Invalid for SFC Mode</td> <td>Invalid for SFC Mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>Filtered Pixels (allow ON/OFF)</td> <td>Filter Pixels Sent to SFC for Scaling</td> </tr> <tr> <td>1</td> <td>0</td> <td>Non-filter (bypass) pixels (allow ON/OFF)</td> <td>Non-Filter Pixels Sent to SFC for Scaling</td> </tr> <tr> <td>1</td> <td>1</td> <td>Non-filter (bypass) pixels (allow ON/OFF) Filtered pixels (always OFF)</td> <td>Filter Pixels Sent to SFC for Scaling.</td> </tr> </tbody> </table>	Pre-Deblock Flag	Post-Deblock Flag	VD Pixels Output to Memory	VD Pixels Output to SFC	0	0	Invalid for SFC Mode	Invalid for SFC Mode	0	1	Filtered Pixels (allow ON/OFF)	Filter Pixels Sent to SFC for Scaling	1	0	Non-filter (bypass) pixels (allow ON/OFF)	Non-Filter Pixels Sent to SFC for Scaling	1	1	Non-filter (bypass) pixels (allow ON/OFF) Filtered pixels (always OFF)	Filter Pixels Sent to SFC for Scaling.	
Pre-Deblock Flag	Post-Deblock Flag	VD Pixels Output to Memory	VD Pixels Output to SFC																				
0	0	Invalid for SFC Mode	Invalid for SFC Mode																				
0	1	Filtered Pixels (allow ON/OFF)	Filter Pixels Sent to SFC for Scaling																				
1	0	Non-filter (bypass) pixels (allow ON/OFF)	Non-Filter Pixels Sent to SFC for Scaling																				
1	1	Non-filter (bypass) pixels (allow ON/OFF) Filtered pixels (always OFF)	Filter Pixels Sent to SFC for Scaling.																				
		<p>VE - image enhanced pixel Output Enable</p> <p>For VE Mode, this field indicates if the VEBOX will enable writing out the image enhanced pixels to memory which is streamed to SFC pipeline for scaling. Filtered data is streamed directly from VEBOX to SFC through a dedicated internal interface. The pixel data send from VE to SFC is YUV format in 12-bit precision irrespective of VEBOX input surface type, pixel precision, chroma format, and color format (RGBA/YUVA). The following table shows allowed usage with VE -image enhanced pixel output enable along with SFC being enabled.</p> <table border="1"> <thead> <tr> <th>VE Output Surface Format</th> <th>Bits per channel</th> <th>Can SFC be enabled ?</th> </tr> </thead> <tbody> <tr> <td>Y8/ NV12/ AYUV/ YUYV/YVYU/UYVY/VYUY</td> <td>8bit</td> <td>Yes</td> </tr> <tr> <td>RGBA8</td> <td>8bit</td> <td>No</td> </tr> <tr> <td>RGBA10</td> <td>10bit</td> <td>No</td> </tr> <tr> <td>RGBA16</td> <td>16bit</td> <td>No</td> </tr> <tr> <td>Y16/ P216/P016/ Y216/ Y416</td> <td>16bit</td> <td>Yes</td> </tr> </tbody> </table> <p>When DN is enabled, DN output is always on.</p> <p>When DI is Enabled, VE must send the first DI constructed surface to SFC in case VEBOX state indicate two DI frames output, while stream out the second DI constructed surface to memory. Else the DI output which is enabled will be sent out to SFC. VE output surface Type is programmed in VE_State command.</p>		VE Output Surface Format	Bits per channel	Can SFC be enabled ?	Y8/ NV12/ AYUV/ YUYV/YVYU/UYVY/VYUY	8bit	Yes	RGBA8	8bit	No	RGBA10	10bit	No	RGBA16	16bit	No	Y16/ P216/P016/ Y216/ Y416	16bit	Yes		
VE Output Surface Format	Bits per channel	Can SFC be enabled ?																					
Y8/ NV12/ AYUV/ YUYV/YVYU/UYVY/VYUY	8bit	Yes																					
RGBA8	8bit	No																					
RGBA10	10bit	No																					
RGBA16	16bit	No																					
Y16/ P216/P016/ Y216/ Y416	16bit	Yes																					
0	VE-SFC Pipe Select																						



SFC_STATE

SFC_STATE			
Source:	BSpec		
Length Bias:	2		
Description			
This command is sent from VDBOX/VEBOX to SFC pipeline at the start of each frame once the lock request is granted.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:23	Media Command Opcode	
		Format:	OpCode
		Value	Name
		Ah	Media Misc [Default]
22:21	SubOpcodeA		
	Default Value:	0h Common	
	Format:	OpCode	
20:16	SubOpcodeB		
	Default Value:	1h SFC_State	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Format:	=n	
	Total Length - 2		
	Value	Name	
	1Eh	Excludes DWord (0,1) [Default]	
1	31:11	Reserved	
		Format:	MBZ



SFC_STATE

10:8

VD/VE Input Ordering Mode

Format:	U3
---------	----

- VD mode: (SFC pipe mode set as "0")
- VE mode: (pipe mode set as "1 and 4")

For values for each mode, please refer to the table below:

Value	Name	Description	Exists If
0		16x16 block z-scan order - no shift	//VD Mode
1		16x16 block z-scan order - 4 pixels shift upward	//VD Mode
2		8x8 block jpeg z-scan order	//VD Mode
3		16x16 block jpeg z-scan order	//VD Mode
4		16x16 block VP8 row-scan order - no shift	//VD Mode
5-7		Reserved	//VD Mode
0		8x4 block column order, 64 pixel column	//VE Mode
1		4x4 block column order, 64 pixel column	//VE Mode
4-7		Reserved	//VE Mode

Programming Notes

This field shall be programmed according to video modes used in VDBOX. NOTE: SFC supports progressive input and output only (Interlaced/MBAFF is not supported).

Video Mode	Surface Format	SFC Input Chroma Sub-Sampling	VD/VE Input Ordering Mode
AVC w/o LF	Monochrome	0	0
AVC w/o LF	420 (NV12)	1	0
AVC with LF	Monochrome	0	1
VP8 w/o LF	420 (NV12)	1	4
JPEG (YUV Interleaved)	Monochrome	0	2
JPEG (YUV Interleaved)	420	1	3
JPEG (YUV Interleaved)	422H_2Y	2	2
JPEG (YUV Interleaved)	422H_4Y	2	3
JPEG (YUV Interleaved)	444	4	2

This field shall be programmed according to Image enhancement modes used in VEBOX.

VEBOX MODE	VEBOX Single Pipe Enable Bit	SFC Input Surface Format	SFC Input Chroma Sub Sampling	VD/VE Input Ordering Mode
1. DN/HP with RGB input 2. Camera pipe (DM) enabled 3. IECP with FECSC, CCM, FGC filters enabled	1	Monochrome	0	1
	1	420 (NV12)	1	1
	1	422H	2	1
	1	444	4	1
All other modes: (Legacy DN/DI/IECP features)	0	Monochrome	0	0
	0	420 (NV12)	1	0
	0	422H	2	0
	0	444	4	0



SFC_STATE

7:4

SFC Input Chroma Sub-Sampling

Value	Name	Description
0	4:0:0	SFC to insert UV channels
1	4:2:0	
2	4:2:2 Horizontal	VD: 2:1:1
3	Reserved	
4	4:4:4 Progressive/Interleaved	
5-6	Reserved	

Programming Notes

This field shall be programmed according to video modes used in VDBOX. NOTE: SFC supports progressive input and output only (Interlaced/MBAFF is not supported).

Video Mode	Surface Format	SFC Input Chroma Sub-Sampling	VD/VE Input Ordering Mode
AVC w/o LF	Monochrome	0	0
AVC w/o LF	420 (NV12)	1	0
AVC with LF	Monochrome	0	1
VP8 w/o LF	420 (NV12)	1	4
JPEG (YUV Interleaved)	Monochrome	0	2
JPEG (YUV Interleaved)	420	1	3
JPEG (YUV Interleaved)	422H_2Y	2	2
JPEG (YUV Interleaved)	422H_4Y	2	3
JPEG (YUV Interleaved)	444	4	2

This field shall be programmed according to Image enhancement modes used in VEBOX.

VEBOX MODE	Surface Format	SFC Input Chroma Sub Sampling	VD/VE Input Ordering Mode
Legacy DN/DI/IECP features	Monochrome	0	0
Legacy DN/DI/IECP features	420 (NV12)	1	0
Legacy DN/DI/IECP features	422H	2	0
Legacy DN/DI/IECP features	444	4	0
Capture/Camera pipe enabled(Demosaic)	Monochrome	0	1
Capture/Camera pipe enabled(Demosaic)	420 (NV12)	1	1
Capture/Camera pipe enabled(Demosaic)	422H	2	1
Capture/Camera pipe enabled(Demosaic)	444	4	1



SFC_STATE																				
	3:0	SFC Pipe Mode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>VD-to-SFC AVS</td> </tr> <tr> <td>1</td> <td></td> <td>VE-to-SFC AVS + IEF + Rotation</td> </tr> <tr> <td>2-3</td> <td></td> <td>Reserved</td> </tr> <tr> <td>4</td> <td></td> <td>VE-to-SFC Integral Image</td> </tr> <tr> <td>5-15</td> <td></td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	0		VD-to-SFC AVS	1		VE-to-SFC AVS + IEF + Rotation	2-3		Reserved	4		VE-to-SFC Integral Image	5-15		Reserved
		Value	Name	Description																
0		VD-to-SFC AVS																		
1		VE-to-SFC AVS + IEF + Rotation																		
2-3		Reserved																		
4		VE-to-SFC Integral Image																		
5-15		Reserved																		
		<p style="text-align: center;">Programming Notes</p> <p>Note: for SFC Pipe mode set to VE-to-SFC AVS mode. IECP pipeline mode MUST be enabled. However, each sub-IECP feature can be turned on/off independently.</p>																		
2	31:28	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ																
	Format:	MBZ																		
	27:16	Input Frame Resolution Height <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U12-1</td> </tr> </table> <p>Minus 1 in unit of pixel [11:0]. It is set to the value of the output resolution or number of pixels streaming into SFC from VD or VEBox. Since the max value support is 4k pixels, the max value allowed is 4K minus 1.</p> <ul style="list-style-type: none"> VDBOX frame height is multiple of 16 for Video source and JPEG formats other than 400, 444 and 422H_2Y. VDBOX frame height is multiple of 8 for JPEG formats 400, 444 and 422H_2Y. VEBOX frame height is multiple of 4. <p><i>Min Resolution</i> is 128 pixels. <i>Max Resolution</i> is upto 4K pixels. e.g. for 1920x1080 content, <i>FrameHeightInMBsMinus1</i> is equal to 1087 (1080 rounded up 16 pixel boundary, minus 1. i.e. effectively specified as 1088 instead).</p>	Format:	U12-1																
	Format:	U12-1																		
		<p style="text-align: center;">Restriction</p> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p>																		
15:12	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">MBZ</td> </tr> </table>	Format:	MBZ																	
Format:	MBZ																			



SFC_STATE																															
	11:0	<p>Input Frame Resolution Width</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U12-1</td> </tr> </table> <p>Minus 1 in unit of pixel [11:0]. It is set to the value of the output resolution or number of pixels streaming into SFC from VD or VEBox. Since the max value support is 4k pixels, the max value allowed is 4K minus 1.</p> <ul style="list-style-type: none"> • VDBOX frame width is multiple of 16 for Video source and JPEG formats other than 400, 444 and 422H_2Y. • VDBOX frame width is multiple of 8 for JPEG formats 400, 444 and 422H_2Y. • VEBOX frame width is multiple of 16. <p><i>Min Resolution</i> is 128 pixels. <i>Max Resolution</i> is upto 4K pixels. e.g. for 1920x1080 content, FrameHeightInMBsMinus1 is equal to 1087 (1080 rounded up 16 pixel boundary, minus 1. i.e. effectively specified as 1088 instead).</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Restriction</td> </tr> </table> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p>	Format:	U12-1	Restriction																										
	Format:	U12-1																													
	Restriction																														
	3	31:17	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																										
Format:	MBZ																														
	16	Reserved																													
	15	Reserved																													
	14:12	<p>Pre-AVS Chroma Downsampling co-siting position Horizontal Direction</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U3</td> </tr> </table> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0/8 (Left full pixel)</td> <td>0 (fraction_in_integer)</td> </tr> <tr> <td>001b</td> <td>1/8</td> <td>1 (fraction_in_integer)</td> </tr> <tr> <td>010b</td> <td>1/4 (2/8)</td> <td>2 (fraction_in_integer)</td> </tr> <tr> <td>011b</td> <td>3/8</td> <td>3 (fraction_in_integer)</td> </tr> <tr> <td>100b</td> <td>1/2 (4/8)</td> <td>4 (fraction_in_integer)</td> </tr> <tr> <td>101b</td> <td>5/8</td> <td>5 (fraction_in_integer)</td> </tr> <tr> <td>110b</td> <td>3/4 (6/8)</td> <td>6 (fraction_in_integer)</td> </tr> <tr> <td>111b</td> <td>7/8</td> <td>7 (fraction_in_integer)</td> </tr> </tbody> </table>	Format:	U3	Value	Name	Description	000b	0/8 (Left full pixel)	0 (fraction_in_integer)	001b	1/8	1 (fraction_in_integer)	010b	1/4 (2/8)	2 (fraction_in_integer)	011b	3/8	3 (fraction_in_integer)	100b	1/2 (4/8)	4 (fraction_in_integer)	101b	5/8	5 (fraction_in_integer)	110b	3/4 (6/8)	6 (fraction_in_integer)	111b	7/8	7 (fraction_in_integer)
	Format:	U3																													
	Value	Name	Description																												
	000b	0/8 (Left full pixel)	0 (fraction_in_integer)																												
001b	1/8	1 (fraction_in_integer)																													
010b	1/4 (2/8)	2 (fraction_in_integer)																													
011b	3/8	3 (fraction_in_integer)																													
100b	1/2 (4/8)	4 (fraction_in_integer)																													
101b	5/8	5 (fraction_in_integer)																													
110b	3/4 (6/8)	6 (fraction_in_integer)																													
111b	7/8	7 (fraction_in_integer)																													
	11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																											
Format:	MBZ																														



SFC_STATE

10:8		<p>Pre-AVS Chroma Downsampling co-siting position Vertical Direction</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U3</td> </tr> </table> <p>This field specifies the fractional position of the bilinear filter for chroma downsampling. In the Y-axis.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr><td>000b</td><td>0/8 (Left full pixel)</td><td>0 (fraction_in_integer)</td></tr> <tr><td>001b</td><td>1/8</td><td>1 (fraction_in_integer)</td></tr> <tr><td>010b</td><td>1/4 (2/8)</td><td>2 (fraction_in_integer)</td></tr> <tr><td>011b</td><td>3/8</td><td>3 (fraction_in_integer)</td></tr> <tr><td>100b</td><td>1/2 (4/8)</td><td>4 (fraction_in_integer)</td></tr> <tr><td>101b</td><td>5/8</td><td>5 (fraction_in_integer)</td></tr> <tr><td>110b</td><td>3/4 (6/8)</td><td>6 (fraction_in_integer)</td></tr> <tr><td>111b</td><td>7/8</td><td>7 (fraction_in_integer)</td></tr> </tbody> </table>	Format:	U3	Value	Name	Description	000b	0/8 (Left full pixel)	0 (fraction_in_integer)	001b	1/8	1 (fraction_in_integer)	010b	1/4 (2/8)	2 (fraction_in_integer)	011b	3/8	3 (fraction_in_integer)	100b	1/2 (4/8)	4 (fraction_in_integer)	101b	5/8	5 (fraction_in_integer)	110b	3/4 (6/8)	6 (fraction_in_integer)	111b	7/8	7 (fraction_in_integer)
Format:	U3																														
Value	Name	Description																													
000b	0/8 (Left full pixel)	0 (fraction_in_integer)																													
001b	1/8	1 (fraction_in_integer)																													
010b	1/4 (2/8)	2 (fraction_in_integer)																													
011b	3/8	3 (fraction_in_integer)																													
100b	1/2 (4/8)	4 (fraction_in_integer)																													
101b	5/8	5 (fraction_in_integer)																													
110b	3/4 (6/8)	6 (fraction_in_integer)																													
111b	7/8	7 (fraction_in_integer)																													
7:6		<p>Pre- AVS Chroma Downsampling Enable</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>This is set to enable chroma downsample bilinear filtering prior to AVS scaling operation. This should be set only if the input chroma format has higher sampling than the chroma format used in the AVS filtering.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 40%;">Value</th> <th style="width: 60%;">Name</th> </tr> </thead> <tbody> <tr><td>00b</td><td>Disable</td></tr> <tr><td>01b</td><td>444 -> 420</td></tr> <tr><td>10b</td><td>444 -> 422</td></tr> <tr><td>11b</td><td>422 -> 420</td></tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>For Integral Image Mode, this field is Reserved and MBZ. The programming of this field should match the input format of VEBox.</p>	Format:	U2	Value	Name	00b	Disable	01b	444 -> 420	10b	444 -> 422	11b	422 -> 420																	
Format:	U2																														
Value	Name																														
00b	Disable																														
01b	444 -> 420																														
10b	444 -> 422																														
11b	422 -> 420																														
5		<p>RGBA_Channel_Swap Enable</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">0</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit should only be used with RGB output formats and CSC conversion is turned on. When this bit is set, the R and B channels are swapped into the output RGB channels as shown in the following table:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 15%;">Bits</th> <th style="width: 20%;">MSB Color Order</th> <th style="width: 50%;">Swapped</th> </tr> </thead> <tbody> <tr><td>RGBA8</td><td>8:8:8:8</td><td>A:B:G:R</td><td>A:R:G:B</td></tr> <tr><td>RGBA10</td><td>2:10:10:10</td><td>A:R:G:B</td><td>A:B:G:R</td></tr> <tr><td>RGB 5:6:5</td><td>5:6:5</td><td>R:G:B</td><td>B:G:R</td></tr> </tbody> </table>	Default Value:	0	Format:	Enable	Name	Bits	MSB Color Order	Swapped	RGBA8	8:8:8:8	A:B:G:R	A:R:G:B	RGBA10	2:10:10:10	A:R:G:B	A:B:G:R	RGB 5:6:5	5:6:5	R:G:B	B:G:R									
Default Value:	0																														
Format:	Enable																														
Name	Bits	MSB Color Order	Swapped																												
RGBA8	8:8:8:8	A:B:G:R	A:R:G:B																												
RGBA10	2:10:10:10	A:R:G:B	A:B:G:R																												
RGB 5:6:5	5:6:5	R:G:B	B:G:R																												



SFC_STATE				
4	4	Reserved		
		Format:	MBZ	
	3:0	Output Surface Format type		
		SFC output surface format type.		
		Reserved		
		Value	Name	Description
		0		AYUV 4:4:4 (8:8:8:8 MSB-A:Y:U:V)
		1		RGBA8 4:4:4:4 (8:8:8:8 MSB-A:B:G:R)
		2		RGBA10 10:10:10:2 (2:10:10:10 MSB-A:R:G:B)
		3		RGB 5:6:5 (5:6:5 MSB-R:G:B)
	4		Planar NV12 4:2:0 8-bit	
	5		Packed YUYV 4:2:2 8-bit	
	6		Packed UYVY 4:2:2 8-bit	
	7		Packed integral Image 32-bit	
	8		Packed integral Image 64-bit	
		Restriction		
		Restriction : For Integral Image Mode, output surface format type must be set to 32/64-bit Integral Image Plane. Driver/SW must ensure the max accumulated integral image value does not exceed the programmable output precision. HW will simply generate wrong value once it overflow in wrap around case.		
4	31:22	Reserved		
		Format:	MBZ	
	21:20	Reserved		
		Format:	MBZ	
	19	CSC Enable		
	Description			
	This field specifies the scaled pixels are converted from YUV to RGB. The output format type must be programmed in one of the RGBAx type.			
	Restriction			
	Restriction : For Integral Image Mode, this field is Reserved and MBZ.			
18	Color Fill Enable			
	Programming Notes			
	This field could be enabled only if the scaled resolution is smaller than the output/display resolution.			



SFC_STATE														
		<p>If enabled, HW will fill the gap with programmable pixel values. Else, nothing will be filled in the gap region.</p> <p>Usage: Color fill must be enabled for the first time/pass when a new surface is allocated/ used. Optional for subsequence frames since the gap region is filled with default pixels by prior passes.</p>												
17:16	Rotation Mode	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U2</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>0 (degrees)</td> </tr> <tr> <td>01b</td> <td>90 Clockwise</td> </tr> <tr> <td>10b</td> <td>180 Clockwise</td> </tr> <tr> <td>11b</td> <td>270 Clockwise</td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 10px;">Programming Notes</p> <p>SFC rotation (90, 180 and 270) should be set only on VEBox input mode and SFC output set to TileY.</p> <p>Restriction:</p> <ul style="list-style-type: none"> For Integral Image Mode, this field is Reserved and MBZ. For VDBox Mode, this field is Reserved and MBZ. For linear or TileX SFC output, this field is Reserved and MBZ. 	Format:	U2	Value	Name	00b	0 (degrees)	01b	90 Clockwise	10b	180 Clockwise	11b	270 Clockwise
Format:	U2													
Value	Name													
00b	0 (degrees)													
01b	90 Clockwise													
10b	180 Clockwise													
11b	270 Clockwise													
15:13	Reserved	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ													
12	Chroma Upsampling Enable	<p>This field enables the high-quality UV channel upsampler prior to IEF filter process. This field should be disabled when the source pixels and output pixels are kept with the same chroma sub-sample type and IEF is disabled.</p> <p style="text-align: center;">Restriction</p> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p>												
11:10	Reserved	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ													
9	Bypass X Adaptive Filtering	<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 20%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Enable X Adaptive Filtering</td> <td></td> </tr> <tr> <td>1</td> <td>Disable X Adaptive Filtering</td> <td>The X direction will use Default Sharpness Level to blend between the smooth and sharp filters rather than the calculated value.</td> </tr> </tbody> </table>	Value	Name	Description	0	Enable X Adaptive Filtering		1	Disable X Adaptive Filtering	The X direction will use Default Sharpness Level to blend between the smooth and sharp filters rather than the calculated value.			
Value	Name	Description												
0	Enable X Adaptive Filtering													
1	Disable X Adaptive Filtering	The X direction will use Default Sharpness Level to blend between the smooth and sharp filters rather than the calculated value.												



SFC_STATE		
8	Bypass Y Adaptive Filtering	
	Value	Name
	0	Enable Y Adaptive Filtering
	1	Disable Y Adaptive Filtering
The Y direction will use Default Sharpness Level to blend between the smooth and sharp filters rather than the calculated value.		
7	AVS Scaling Enable	
	Value	Name
	1	Enable
	0	Disable
The scaling factor is ignored and a scaling ratio of 1:1 is assumed.		
6	Adaptive Filter for all Channels	
	Value	Name
	1	Enable Adaptive Filter on UV/RB Channels
	0	Disable Adaptive Filter on UV/RB Channels
8-tap Adaptive Filter Mode is on		
5:4	AVS Filter Mode	
	Value	Name
	0	5x5 Poly-phase filter + Bilinear (adaptive)
	1	8x8 poly-phase filter + Bilinear (adaptive)
	2	Bilinear filter only
	3	Reserved
Programming Notes		
In VD-to-SFC mode, value of 1 is not allowed.		
3	Reserved	
	Format:	MBZ
2	IEF4Smooth_Enable	
	Value	Name
	0	[Default]
	1	
IEF is operating as a content adaptive detail filter based on 5x5 region.		
IEF is operating as a content adaptive smooth filter based on 3x3 region		
Restriction		
Restriction : For Integral Image Mode, this field is Reserved and MBZ.		
1	Skin Tone Tuned IEF_Enable	
	Exists If:	//IEF Enable = 1
Restriction		
Restriction : For Integral Image Mode, this field is Reserved and MBZ.		



SFC_STATE		
0	IEF Enable	
	Value	Name
	1	Enable
	0	Disable
	Restriction	
Restriction : For Integral Image Mode and VD Mode, this field is Reserved and MBZ.		
5	Reserved	
	Format:	MBZ
	Source Region Height	
	Format:	U12-1
	Source/Crop Region Height Minus 1 of the Input Frame in Unit of Pixel [11:0].	
	This field specifies the source/crop region of the input frame used for scaling of the graphic view. It defines the out-of-frame boundary used prior to AVS/IEF interpolation operation. The max value should be programmed to be equal or small than the input FrameHeightInMBsMinus1 field. e.g. for 1920x1080 content, FrameHeightInMBsMinus1 is equal to1087 (1088 lines); however, the crop region height should be set to 1079(1080 lines). The last 8 lines are assumed to be not usable and should not be used as source pixels for Scaling or IEF operations. Otherwise, the bad pixels will breach and cause artifacts into the scaled output frame.	
	Restriction	
	Restriction : For Integral Image Mode, this field is Reserved and MBZ.	
	Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions.	
	Min Resolution is 128 pixels. Max Resolution is 4K pixels.	
15:12	Reserved	
	Format:	MBZ
11:0	Source Region Width	
	Format:	U12-1
	Source/Crop Region Width Minus 1 of the Input Frame in Unit of Pixel [11:0].	
This field specifies the source/crop region of the input frame used for scaling of the graphic view. It defines the out-of-frame boundary used prior to AVS/IEF interpolation operation. The max value should be programmed to be equal or small than the input FrameWidthInMBsMinus1 field. e.g. for 1920x1080 content, FrameWidthInMBsMinus1 is equal to1919 (1920 pixel wide); however, the crop region width should be set to less than 1909(1910 pixel wide). The last 10 pixels of the frame are assumed to be not usable and should not be used as source pixels for Scaling or IEF operations. Otherwise, the bad pixels will breach and cause artifacts into the scaled output frame.		
Restriction		
Restriction : For Integral Image Mode, this field is Reserved and MBZ.		



SFC_STATE						
	<p>Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions. Min Resolution is 128 pixels. Max Resolution is 4K pixels.</p>					
6	<p>31:28 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ				
	<p>27:16 Source Region Vertical Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U12</td> </tr> </table> <p>Vertical Offset Of The SRC Region Relative To The Starting Position Of The Input Frame In Unit Of Pixel [11:0]</p> <p>This field specifies the vertical offset of the starting position of the scaled region relatives to the starting position (pixel 0,0) of the output frame. It defines the out-of-frame boundary used prior to AVS/IEF interpolation operation. This value should be set to zero if the starting corner of the crop region is same as the input frame region. The sum of this value and the src/crop region size heightminus1 must be programmed to be equal or small than the input FrameHeightinMBminus 1 field.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Restriction</th> </tr> <tr> <td>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> <tr> <td>Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions.</td> </tr> </table>	Format:	U12	Restriction	Restriction : For Integral Image Mode, this field is Reserved and MBZ.	Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions.
	Format:	U12				
	Restriction					
Restriction : For Integral Image Mode, this field is Reserved and MBZ.						
Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions.						
<p>15:12 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ					
<p>11:0 Source Region Horizontal Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U12</td> </tr> </table> <p>Horizontal Offset Of The SRC Region Relative To The Starting Position Of The Input Frame In Unit Of Pixel [11:0]</p> <p>This field specifies the horizontal offset of the starting position of the scaled region relatives to the starting position (pixel 0,0) of the output frame. It defines the out-of-frame boundary used prior to AVS/IEF interpolation operation. This value should be set to zero if the starting corner of the crop region is same as the input frame region. The sum of this value and the src/crop region size widthminus1 must be programmed to be equal or small than the input FrameWidthinMBminus 1 field.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Restriction</th> </tr> <tr> <td>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> <tr> <td>Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions.</td> </tr> </table>	Format:	U12	Restriction	Restriction : For Integral Image Mode, this field is Reserved and MBZ.	Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions.	
Format:	U12					
Restriction						
Restriction : For Integral Image Mode, this field is Reserved and MBZ.						
Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions.						



SFC_STATE					
7	31:28	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ	
	Format:	MBZ			
	27:16	Output Frame Height <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U12-1</td> </tr> </table> <p>It is set to the value of the final output resolution of the graphic view. Since the max value support is 4k pixels, the max value allowed is 4K minus 1.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Restriction</td> </tr> </table> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p> <p>Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions. Min Resolution is 128 pixels. Max Resolution is 4K pixels.</p>	Format:	U12-1	Restriction
	Format:	U12-1			
Restriction					
15:12	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ				
11:0	Output Frame Width <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U12-1</td> </tr> </table> <p>It is set to the value of the final output resolution of the graphic view. Since the max value support is 4k pixels, the max value allowed is 4K minus 1.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Restriction</td> </tr> </table> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p> <p>Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions. Min Resolution is 128 pixels. Max Resolution is 4K pixels.</p>	Format:	U12-1	Restriction	
Format:	U12-1				
Restriction					
8	31:28	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ	
	Format:	MBZ			
	27:16	Scaled Region Size Height <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">U12-1 Multiple of 2 Pixels</td> </tr> </table> <p>It is set to the height of the scaled region over the output frame of the graphic view.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>The Max Value = < [The Output Frame Height Minus1].</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Restriction</td> </tr> </table> <p>Restriction : For AVS mode, if rotation_mode = 0/180, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions.</p>	Format:	U12-1 Multiple of 2 Pixels	Programming Notes
Format:	U12-1 Multiple of 2 Pixels				
Programming Notes					
Restriction					



SFC_STATE			
	<p>For AVS mode, if rotation_mode = 90/270, the restriction is tied to chroma output format type: 420/422 - multiple of 2. 444/400 - no restrictions. Min Resolution is 128 pixels. Max Resolution is 4K pixels.</p>		
15:12	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
11:0	<p>Scaled Region Size Width</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U12-1 Multiple of 2 Pixels</td> </tr> </table> <p>It is set to the Width of the scaled region over the output frame of the graphic view.</p> <p style="text-align: center;">Programming Notes</p> <p>The Max Value =< [The Output Frame Width Minus1].</p> <p style="text-align: center;">Restriction</p> <p>Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions. Min Resolution is 128 pixels. Max Resolution is 4K pixels.</p>	Format:	U12-1 Multiple of 2 Pixels
Format:	U12-1 Multiple of 2 Pixels		
9	<p>31:29 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ	
	<p>28:16 Scaled Region Vertical Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>S12</td> </tr> </table> <p>Vertical Offset (in pixels) Of The Scaled Region Relatives to The Starting Position Of The Output Frame In Unit Of Pixel [11:0]</p> <p>This field specifies the vertical offset of the starting position of the scaled region relatives to the starting position (pixel 0,0) of the output frame. The gap between the scaled and output frame shall be filled by hardware with a set of programmed YUV/RGB values (Grey Bar). This value should be set to zero if the starting corner of the scaled region is same as the output frame region. The sum of this value and the scaled region size Heightminus1 must be programmed to be equal or small than the output FrameHeightinMBminus 1 field plus 16.</p> <p style="text-align: center;">Restriction</p> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ. Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions. Restriction : This field must be set to zero if SFC Ouput surface format type is NV12.</p>	Format:	S12
	Format:	S12	
<p>15:13 Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ		
<p>12:0 Scaled Region Horizontal Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>S12</td> </tr> </table>	Format:	S12	
Format:	S12		



SFC_STATE									
	<p>Horizontal Offset (in pixels) Of The Scaled Region Relatives to The Starting Position Of The Output Frame In Unit Of Pixel [11:0]</p> <p>This field specifies the horizontal offset of the starting position of the scaled region relatives to the starting position (pixel 0,0) of the output frame. The gap between the scaled and output frame shall be filled by hardware with a set of programmed YUV/RGB values (Grey Bar). This value should be set to zero if the starting corner of the scaled region is same as the output frame region. The sum of this value and the scaled region size Widthminus1 must be programmed to be equal or small than the output FrameWidthinMBminus 1 field plus 16.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Restriction</th> </tr> <tr> <td colspan="2">Restriction : For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> <tr> <td colspan="2">Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions.</td> </tr> <tr> <td colspan="2">Restriction : This field must be set to zero if SFC Ouput surface format type is NV12.</td> </tr> </table>	Restriction		Restriction : For Integral Image Mode, this field is Reserved and MBZ.		Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions.		Restriction : This field must be set to zero if SFC Ouput surface format type is NV12.	
Restriction									
Restriction : For Integral Image Mode, this field is Reserved and MBZ.									
Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions.									
Restriction : This field must be set to zero if SFC Ouput surface format type is NV12.									
10	31:26	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
	25:16	<p>Gray Bar Pixel - Y/R</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="text-align: center;">10-bit UNORM Type</td> </tr> </table> <p>Range:[0.0, +1.0]</p> <p>This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in Y or R channel on the AYUV or RGBA domain respectively.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Restriction</th> </tr> <tr> <td colspan="2">Restriction : For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	Format:	10-bit UNORM Type	Restriction		Restriction : For Integral Image Mode, this field is Reserved and MBZ.		
	Format:	10-bit UNORM Type							
Restriction									
Restriction : For Integral Image Mode, this field is Reserved and MBZ.									
15:10	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ								
9:0	<p>Gray Bar Pixel - U/G</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="text-align: center;">10-bit UNORM Type</td> </tr> </table> <p>Range:[0.0, +1.0]</p> <p>This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in U or G channel on the AYUV or RGBA domain respectively.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Restriction</th> </tr> <tr> <td colspan="2">Restriction : For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	Format:	10-bit UNORM Type	Restriction		Restriction : For Integral Image Mode, this field is Reserved and MBZ.			
Format:	10-bit UNORM Type								
Restriction									
Restriction : For Integral Image Mode, this field is Reserved and MBZ.									
11	31:26	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ								



SFC_STATE															
	<table border="1"> <tr> <td style="width: 10%; text-align: right;">25:16</td> <td>Gray Bar Pixel - V/B</td> </tr> <tr> <td>Format:</td> <td>10-bit UNORM Type</td> </tr> <tr> <td colspan="2">Range:[0.0, +1.0]</td> </tr> <tr> <td colspan="2">This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in V or B channel on the AYUV or RGBA domain respectively.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Restriction</td> </tr> <tr> <td colspan="2">Restriction : For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	25:16	Gray Bar Pixel - V/B	Format:	10-bit UNORM Type	Range: [0.0, +1.0]		This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in V or B channel on the AYUV or RGBA domain respectively.		Restriction		Restriction : For Integral Image Mode, this field is Reserved and MBZ.			
	25:16	Gray Bar Pixel - V/B													
	Format:	10-bit UNORM Type													
	Range: [0.0, +1.0]														
This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in V or B channel on the AYUV or RGBA domain respectively.															
Restriction															
Restriction : For Integral Image Mode, this field is Reserved and MBZ.															
<table border="1"> <tr> <td style="width: 10%; text-align: right;">15:10</td> <td>Reserved</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	15:10	Reserved	Format:	MBZ											
15:10	Reserved														
Format:	MBZ														
<table border="1"> <tr> <td style="width: 10%; text-align: right;">9:0</td> <td>Gray Bar Pixel - A</td> </tr> <tr> <td>Format:</td> <td>10-bit UNORM Type</td> </tr> <tr> <td colspan="2">Range:[0.0, +1.0]</td> </tr> <tr> <td colspan="2">This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in A channel on the AYUV or RGBA domain respectively.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Restriction</td> </tr> <tr> <td colspan="2">Restriction : For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	9:0	Gray Bar Pixel - A	Format:	10-bit UNORM Type	Range: [0.0, +1.0]		This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in A channel on the AYUV or RGBA domain respectively.		Restriction		Restriction : For Integral Image Mode, this field is Reserved and MBZ.				
9:0	Gray Bar Pixel - A														
Format:	10-bit UNORM Type														
Range: [0.0, +1.0]															
This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in A channel on the AYUV or RGBA domain respectively.															
Restriction															
Restriction : For Integral Image Mode, this field is Reserved and MBZ.															
<table border="1"> <tr> <td style="width: 10%; text-align: right;">31:26</td> <td>Reserved</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	31:26	Reserved	Format:	MBZ											
31:26	Reserved														
Format:	MBZ														
12	<table border="1"> <tr> <td style="width: 10%; text-align: right;">25:16</td> <td>UV Default value for V channel (For Mono Input Support)</td> </tr> <tr> <td>Exists If:</td> <td>//Input NOT originated by VEBOX.</td> </tr> <tr> <td>Format:</td> <td>10-bit UNORM Type</td> </tr> <tr> <td colspan="2">Range:[0.0, +1.0]</td> </tr> <tr> <td colspan="2">This field specifies the UV default value fill in to the UV output channels when input from VDBOX is set to Monochrome.</td> </tr> <tr> <td colspan="2" style="text-align: center;">Restriction</td> </tr> <tr> <td colspan="2">Restriction : Not used when input is originated by VEBOX (Including Integral Image Mode).</td> </tr> </table>	25:16	UV Default value for V channel (For Mono Input Support)	Exists If:	//Input NOT originated by VEBOX.	Format:	10-bit UNORM Type	Range: [0.0, +1.0]		This field specifies the UV default value fill in to the UV output channels when input from VDBOX is set to Monochrome.		Restriction		Restriction : Not used when input is originated by VEBOX (Including Integral Image Mode).	
25:16	UV Default value for V channel (For Mono Input Support)														
Exists If:	//Input NOT originated by VEBOX.														
Format:	10-bit UNORM Type														
Range: [0.0, +1.0]															
This field specifies the UV default value fill in to the UV output channels when input from VDBOX is set to Monochrome.															
Restriction															
Restriction : Not used when input is originated by VEBOX (Including Integral Image Mode).															
	<table border="1"> <tr> <td style="width: 10%; text-align: right;">15:10</td> <td>Reserved</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	15:10	Reserved	Format:	MBZ										
15:10	Reserved														
Format:	MBZ														
	<table border="1"> <tr> <td style="width: 10%; text-align: right;">9:0</td> <td>UV Default value for U channel (For Mono Input Support)</td> </tr> <tr> <td>Exists If:</td> <td>//Input NOT originated by VEBOX.</td> </tr> <tr> <td>Format:</td> <td>10-bit UNORM Type</td> </tr> </table>	9:0	UV Default value for U channel (For Mono Input Support)	Exists If:	//Input NOT originated by VEBOX.	Format:	10-bit UNORM Type								
9:0	UV Default value for U channel (For Mono Input Support)														
Exists If:	//Input NOT originated by VEBOX.														
Format:	10-bit UNORM Type														



SFC_STATE						
	<p>Range:[0.0, +1.0]</p> <p>This field specifies the UV default value fill in to the UV output channels when input from VDBOX is set to Monochrome.</p> <p style="text-align: center;">Restriction</p> <p>Restriction : Not used when input is originated by VEBOX (Including Integral Image Mode).</p>					
13	<p>31:10 Reserved</p> <p>Format: MBZ</p>					
	<p>9:0 Alpha Default Value</p> <p>Format: 10-bit UNORM Type</p> <p>Range:[0.0, +1.0]</p> <p>This field specifies the Alpha default value fill into the alpha output channel when output format type is set to RGBA8/10.</p> <p style="text-align: center;">Restriction</p> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p>					
14	<p>31:21 Reserved</p> <p>Format: MBZ</p>					
	<p>20:0 Scaling Factor Height</p> <p>Format: U4.17</p> <p>This field specifies the scaling ratio of the vertical sizes between the crop/source region and the scaled region. The destination pixel coordinate, y-axis, is multiplied with this scaling factor to mapping back to the source input pixel coordinate.</p> <p>The field specifies the ratio of crop height resolution/ scaled height resolution. This implies $1/sf_u$ in the equation.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>Reserved</td> </tr> </tbody> </table> <p style="text-align: center;">Restriction</p> <p>Restriction : Integral Image Mode supports downsampling only.</p>	Value	Name	Description	0	
Value	Name	Description				
0		Reserved				
15	<p>31:21 Reserved</p> <p>Format: MBZ</p>					
	<p>20:0 Scaling Factor Width</p> <p>Format: U4.17</p> <p>This field specifies the scaling ratio of the horizontal sizes between the crop/source region and the scaled region. The destination pixel coordinate, x-axis, is multiplied with this scaling factor to mapping back to the source input pixel coordinate.</p> <p>The field specifies the ratio of crop width resolution/ scaled width resolution. This implies $1/sf_u$ in</p>					



SFC_STATE																			
		<p>the equations above.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 35%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	0		Reserved											
Value	Name	Description																	
0		Reserved																	
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; color: #0070C0;">Restriction</th> </tr> </thead> <tbody> <tr> <td>Restriction : Integral Image Mode supports downsampling only.</td> </tr> </tbody> </table>	Restriction	Restriction : Integral Image Mode supports downsampling only.															
Restriction																			
Restriction : Integral Image Mode supports downsampling only.																			
16	31:0	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																		
17	31:12	<p>Output Frame Surface Base Address</p> <p>Specifies the 4K byte aligned frame buffer address for outputting the scaled up/down image. Data is stored in Tile-Y format.</p> <p>For Integral Image mode, the accumulated integral image values will be packed linear in this surface.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>This field is ignored if I-frame only mode is set to 0 (Disable).</td> </tr> </tbody> </table>	Programming Notes	This field is ignored if I-frame only mode is set to 0 (Disable).															
Programming Notes																			
This field is ignored if I-frame only mode is set to 0 (Disable).																			
	11:0	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																		
18	31:16	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																		
	15:0	<p>Output Frame Surface Base Address High</p> <p>This field is for the upper range [47:32] of Output Frame Surface Base Address.</p> <p>For Integral Image mode, the accumulated integral image values will be packed linear in this surface.</p>																	
19	31:15	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																		
	14:13	<p>Output Surface Tiled Mode</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 35%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td style="text-align: center;">2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td style="text-align: center;">3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
Format:	U2																		
Value	Name	Description																	
0h	TRMODE_NONE	No tiled resource																	
1h	TRMODE_TILEYF	4KB tiled resources																	
2h	TRMODE_TILEYS	64KB tiled resources																	
3h	Reserved																		



SFC_STATE		
12	Output Frame Surface Base Address - Row Store Scratch Buffer Cache Select	
	Format:	MBZ
	Value	Name
	0	Disable [Default]
	This field must be programmed to 0	
Programming Notes		
This must be set to 0		
11	Reserved	
	Format:	MBZ
10	Output Frame Surface Base Address - Memory Compression Mode	
	Format:	U1
	Description	
	Distinguishes vertical from horizontal compression. Please refer to vol1a Memory Data Formats chapter - section media Memory Compression for more details.	
	Value	Name
0	Vertical Compression	
1	Horizontal Compression	
9	Output Frame Surface Base Address - Memory Compression Enable	
	Format:	Enable
Memory compression will be attempted for this surface.		
8:7	Output Frame Surface Base Address - Arbitration Priority Control	
	Format:	HEVC_ARBITRATION_PRIORITY
6:1	Output Frame Surface Base Address - Index to Memory Object Control State (MOCS) Tables	
	Format:	U6
	The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.	
	The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
0	Reserved	



SFC_STATE																			
20	31:12	<p>AVS Line Buffer Surface Base Address</p> <p>Specifies the 4K byte aligned frame buffer address for scratch space used for row/column store. This surface is used only if the internal buffer inside the SFC HW is not large enough to contain all row/column memory accesses. The AVS line buffer needs to be a valid address even for 1:1 scaling if SFC is used.</p>																	
	11:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																		
21	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ															
	Format:	MBZ																	
15:0	<p>AVS Line Buffer Surface Base Address High</p> <p>This field is for the upper range [47:32] of AVS Line Buffer Surface Base Address. AVS Line buffer address needs to be valid even for 1:1 scaling if SFC is used.</p>																		
22	31:15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ															
	Format:	MBZ																	
	14:13	<p>AVS Line Buffer Tiled Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Format:	U2																	
Value	Name	Description																	
0h	TRMODE_NONE	No tiled resource																	
1h	TRMODE_TILEYF	4KB tiled resources																	
2h	TRMODE_TILEYS	64KB tiled resources																	
3h	Reserved																		
12	<p>AVS Line Buffer Base Address - Row Store Scratch Buffer Cache Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LLC [Default]</td> <td>Buffer going to LLC</td> </tr> </tbody> </table> <table border="1" style="width: 100%; text-align: center; background-color: #e6f2ff;"> <tr> <td colspan="3">Programming Notes</td> </tr> </table> <p>This surface does not support to put in Row Store Scratch Buffer. Must be set to 0</p>	Format:	U1	Value	Name	Description	0	LLC [Default]	Buffer going to LLC	Programming Notes									
Format:	U1																		
Value	Name	Description																	
0	LLC [Default]	Buffer going to LLC																	
Programming Notes																			
11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																		



SFC_STATE								
	10	<p>AVS Line Buffer Base Address - Memory Compression Mode</p> <table border="1"> <tr> <td>Default Value:</td> <td>0 Horizontal Compression Mode</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Distinguishes vertical from horizontal compression. Please refer to vol1a Memory Data Formats chapter - section media Memory Compression for more details.</p> <table border="1"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> <tr> <td>Memory compression is not supported. This bit is not used. Default to 0</td> </tr> </table>	Default Value:	0 Horizontal Compression Mode	Format:	U1	Programming Notes	Memory compression is not supported. This bit is not used. Default to 0
	Default Value:	0 Horizontal Compression Mode						
	Format:	U1						
	Programming Notes							
	Memory compression is not supported. This bit is not used. Default to 0							
9	<p>AVS Line Buffer Base Address - Memory Compression Enable</p> <table border="1"> <tr> <td>Default Value:</td> <td>0 Disable</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit control memory compression for this surface</p> <table border="1"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> <tr> <td>This bit must be set to 0 (Memory compression is not supported in this surface)</td> </tr> </table>	Default Value:	0 Disable	Format:	Enable	Programming Notes	This bit must be set to 0 (Memory compression is not supported in this surface)	
Default Value:	0 Disable							
Format:	Enable							
Programming Notes								
This bit must be set to 0 (Memory compression is not supported in this surface)								
8:7	<p>AVS Line Buffer Base Address - Arbitration Priority Control</p> <table border="1"> <tr> <td>Format:</td> <td>HEVC_ARBITRATION_PRIORITY</td> </tr> </table>	Format:	HEVC_ARBITRATION_PRIORITY					
Format:	HEVC_ARBITRATION_PRIORITY							
6:1	<p>AVS Line Buffer Base Address - Index to Memory Object Control State (MOCS) Tables</p> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.</p> <p>The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>	Format:	U6					
Format:	U6							
0	Reserved							
23	31:12	<p>IEF Line Buffer Surface Base Address</p> <p>Specifies the 4K byte aligned frame buffer address for the scratch space used for row/column store.</p> <p>This surface is used only if the internal buffer inside SFC HW is not large enough to contain all row/column memory accesses.</p> <table border="1"> <tr> <td style="text-align: center;">Restriction</td> </tr> <tr> <td>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	Restriction	Restriction : For Integral Image Mode, this field is Reserved and MBZ.				
	Restriction							
Restriction : For Integral Image Mode, this field is Reserved and MBZ.								
11:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ							
24	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ						



SFC_STATE																			
	15:0	IEF Line Buffer Surface Base Address High This field is for the upper range [47:32] of IEF Line Buffer Surface Base Address.																	
		<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Restriction</th> </tr> </thead> <tbody> <tr> <td colspan="2">Restriction : For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </tbody> </table>	Restriction		Restriction : For Integral Image Mode, this field is Reserved and MBZ.														
Restriction																			
Restriction : For Integral Image Mode, this field is Reserved and MBZ.																			
25	31:15	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ															
	Format:	MBZ																	
	14:13	IEF Line Buffer Tiled Mode <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>For Media Surfaces: This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
	Format:	U2																	
	Value	Name	Description																
0h	TRMODE_NONE	No tiled resource																	
1h	TRMODE_TILEYF	4KB tiled resources																	
2h	TRMODE_TILEYS	64KB tiled resources																	
3h	Reserved																		
12	IEF Line Buffer Base Address - Row Store Scratch Buffer Cache Select <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LLC [Default]</td> <td>Buffer going to LLC</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	0	LLC [Default]	Buffer going to LLC	Programming Notes		This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0							
Format:	U1																		
Value	Name	Description																	
0	LLC [Default]	Buffer going to LLC																	
Programming Notes																			
This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0																			
11	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																
Format:	MBZ																		
10	IEF Line Buffer Base Address - Memory Compression Mode <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Distinguishes vertical from horizontal compression. Please refer to vol1a Memory Data Formats chapter - section media Memory Compression for more details.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">Must be zero; memory compression is not supported for this surface. Default to0</td> </tr> </tbody> </table>	Default Value:	0	Format:	U1	Programming Notes		Must be zero; memory compression is not supported for this surface. Default to0											
Default Value:	0																		
Format:	U1																		
Programming Notes																			
Must be zero; memory compression is not supported for this surface. Default to0																			
9	IEF Line Buffer Base Address - Memory Compression Enable <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0 Disable</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table>	Default Value:	0 Disable	Format:	Enable														
	Default Value:	0 Disable																	
Format:	Enable																		



SFC_STATE			
		Programming Notes	
		Memory compression is not supported for this surface Must be 0.	
	8:7	IEF Line Buffer Base Address - Arbitration Priority Control	
		Format: HEVC_ARBITRATION_PRIORITY	
	6:1	IEF Line Buffer Base Address - Index to Memory Object Control State (MOCS) Tables	
		Format: U6	
		The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.	
		The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
	0	Reserved	
26	31:0	Reserved	
		Format: MBZ	
27	31:0	Reserved	
		Format: MBZ	
28	31:0	Reserved	
		Format: MBZ	
29	31:28	Output Surface Format	
	27	Output Surface Interleave Chroma Enable	
	26:20	Reserved	
		Format: MBZ	
19:3	Output Surface Pitch		
		Format: U17-1 Pitch in (Bytes - 1)	
	This field specifies the surface pitch.		
	Value	Name	Description
	[0,2047]	SURFTYPE_BUFFER Surfaces	[1B, 2048B]
	[0, 524287]	Other Linear Surfaces	[64B, 512KB] = [1 CL, 8K CLs]
	[511, 524287]	X-tiled Surface	[512B, 256KB] = [1tile, 512 tiles]
	[127, 524287]	Y-tiled surfaces	[128B,256KB] = [1 tile, 2048 tiles]
	Programming Notes		
	<ul style="list-style-type: none"> • For tiled surfaces, the pitch must be a multiple of the tile width • For Linear surfaces, the pitch must be a multiple of CL (64B) width • If Half Pitch for Chroma is set, this field must be a multiple of two tile widths for tiled 		



SFC_STATE

surfaces, or a multiple of 2 bytes for linear surfaces.

If Media Memory Compression is enabled, the following max pitch size restriction must be honored. For larger resolution, Media Memory compression Must be disabled.

Tiling Mode	Pixel Format	Max Frame Width (bytes)	Max Frame Width (pixels)	Max Pitch (bytes)
Legacy 4K	8bpp	16k	16k	16k + 127
	16bpp	16k	8k	16k + 127
	32bpp	16k	4k	16k + 127
	64bpp	16k	2k	16k + 127
	128bpp	16k	1k	16k + 127
TileYF	8bpp	8k	8k	8k + 63
	16bpp	16k	8k	16k + 127
	32bpp	16k	4k	16k + 127
	64bpp	16k	2k	16k + 255
	128bpp	16k	1k	16k + 255
TileYS	8bpp	16k	16k	16k + 255
	16bpp	16k	8k	16k + 511
	32bpp	16k	4k	16k + 511
	64bpp	16k	2k	16k + 1023
	128bpp	16k	1k	16k + 1023

2 Output Surface Half Pitch For Chroma

Exists If:	//PLANAR Surface Formats Only
Format:	Enable

This field indicates that the chroma plane(s) will use a pitch equal to half the value specified in the Surface Pitch field.

1 Output Surface Tiled

Format:	Boolean
---------	---------

This field specifies whether the surface is tiled.

Value	Name	Description
1	True	Tiled
0	FALSE	Linear

Programming Notes

- Linear surfaces can be mapped to Main Memory (uncached) or System Memory (cacheable, snooped). Tiled surfaces can only be mapped to Main Memory.



SFC_STATE													
	<ul style="list-style-type: none"> The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this bit. The tiled surfaces of current picture and reference picture should be declared as the identical type in VDI mode with the identical Height, Width and Format. 												
0	<p>Output Surface Tile Walk</p> <table border="1"> <tr> <td>Format:</td> <td>SFC_Tile_Walk</td> </tr> </table> <p>This field specifies the type of memory tiling (XMajor or YMajor) employed to tile this surface. See <i>Memory Interface Functions</i> for details on memory tiling and restrictions.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>TILEWALK_XMAJOR</td> </tr> <tr> <td style="text-align: center;">1</td> <td>TILEWALK_YMAJOR</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this bit. <p>This field is ignored when the surface is linear.</p>	Format:	SFC_Tile_Walk	Value	Name	0	TILEWALK_XMAJOR	1	TILEWALK_YMAJOR				
Format:	SFC_Tile_Walk												
Value	Name												
0	TILEWALK_XMAJOR												
1	TILEWALK_YMAJOR												
30	<p>31:30 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>29:16 Output Surface X Offset For U</p> <table border="1"> <tr> <td>Exists If:</td> <td>//PLANAR Surface Formats Only</td> </tr> <tr> <td>Format:</td> <td>U14 Pixel Offset</td> </tr> </table> <p>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.</p> <p style="text-align: center;">Programming Notes</p> <p>For PLANAR_420 and PLANAR_422 surface formats, this field must be zero.</p> <p>15:14 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>13:0 Output Surface Y Offset For U</p> <table border="1"> <tr> <td>Exists If:</td> <td>//PLANAR Surface Formats Only</td> </tr> <tr> <td>Format:</td> <td>U14 Pixel Row Offset</td> </tr> </table> <p>This field specifies the vertical offset in rows from the Surface Base Address to the start (origin) of the U(Cb) plane or the interleaved UV plane if Interleave Chroma is enabled.</p> <p style="text-align: center;">Programming Notes</p> <p>For PLANAR_420 and PLANAR_422 surface formats, this field must be multiple of 16 pixels - i.e. multiple MBs.</p>	Format:	MBZ	Exists If:	//PLANAR Surface Formats Only	Format:	U14 Pixel Offset	Format:	MBZ	Exists If:	//PLANAR Surface Formats Only	Format:	U14 Pixel Row Offset
Format:	MBZ												
Exists If:	//PLANAR Surface Formats Only												
Format:	U14 Pixel Offset												
Format:	MBZ												
Exists If:	//PLANAR Surface Formats Only												
Format:	U14 Pixel Row Offset												
31	<p>31:30 Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												



SFC_STATE								
	29:16	<p>Output Surface X Offset For V</p> <table border="1"> <tr> <td>Exists If:</td> <td>//PLANAR Surface Formats with Interleaved Chroma Disable</td> </tr> <tr> <td>Format:</td> <td>U14 Pixel Offset</td> </tr> </table> <p>This field specifies the horizontal offset in pixels from the Surface Base Address to the start (origin) of the V(Cr) plane.</p> <table border="1" style="background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table> <p>For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.</p>	Exists If:	//PLANAR Surface Formats with Interleaved Chroma Disable	Format:	U14 Pixel Offset	Programming Notes	
	Exists If:	//PLANAR Surface Formats with Interleaved Chroma Disable						
	Format:	U14 Pixel Offset						
	Programming Notes							
15:14	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ							
	13:0	<p>Output Surface Y Offset For V</p> <table border="1"> <tr> <td>Exists If:</td> <td>//PLANAR Surface Formats with Interleaved Chroma Disable</td> </tr> <tr> <td>Format:</td> <td>U14 Pixel Offset</td> </tr> </table> <p>This field specifies the vertical offset in rows from the Surface Base Address to the start (origin) of the V(Cr) plane.</p> <table border="1" style="background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table> <p>For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.</p>	Exists If:	//PLANAR Surface Formats with Interleaved Chroma Disable	Format:	U14 Pixel Offset	Programming Notes	
	Exists If:	//PLANAR Surface Formats with Interleaved Chroma Disable						
	Format:	U14 Pixel Offset						
	Programming Notes							
32	31:1	Reserved						
33	31:0	Reserved						



Shift Left

shl - Shift Left		
Source:	Eulsa	
Length Bias:	4	
Description		
<p>Perform component-wise logical left shift of the bits in src0 by the shift count indicated in src1, storing the results in dst, inserting zero bits in the number of LSBs indicated by the shift count.</p> <p>Hardware detects overflow properly and uses it to perform any saturation operation on the result, as long as the shifted result is within 33 bits. Otherwise, the result is undefined.</p> <p>Note: For word and DWord operands, the accumulators have 33 bits.</p> <p>In QWord mode, the shift count is taken from the low six bits of src1 regardless of the src1 type and treated as an unsigned integer in the range 0 to 63. Otherwise 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. The operation uses QWord mode if src0 or dst has the Q or UQ type but not if src1 is the only operand with the Q or UQ type.</p>		
<p>Format:</p> <p>[(pred)] shl[.cmod] (exec_size) dst src0 src1</p>		
Restriction		
Restriction : Accumulator cannot be destination, implicit or explicit.		
Syntax		
[(pred)] shl[.cmod] (exec_size) reg reg reg [(pred)] shl[.cmod] (exec_size) reg reg imm32		
Pseudocode		
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { shiftCnt = src0 or dst has Q or UQ type ? src1.chan[n] & 0x3F : src1.chan[n] & 0x1F dst.chan[n] = src0.chan[n] << shiftCnt; } }</pre>		
Predication	Conditional Modifier	Source Modifier
Y	Y	Y
Src Types		Dst Types
*B,*W,*D		*B,*W,*D
*W,*D		*W,*D
*W,*D		*Q
*Q		*W,*D
*Q		*Q
DWord	Bit	Description
0..3	127:64	RegSource



shl - Shift Left			
		Exists If:	(([RegSource][Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG
	127:64	ImmSource	
		Exists If:	(([ImmSource][Src1.RegFile]='IMM')
	63:32	Format:	EU_INSTRUCTION_SOURCES_REG_IMM
		Operand Controls	
	31:0	Format:	EU_INSTRUCTION_OPERAND_CONTROLS
		Header	
		Format:	EU_INSTRUCTION_HEADER



Shift Right

shr - Shift Right			
Source:	Eulsa		
Length Bias:	4		
Description			
<p>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 can have different types and can be signed or unsigned. Note: For word and DWord operands, the accumulators have 33 bits. Note: For unsigned src0 types, shr and asr produce the same result.</p> <p>In QWord mode, the shift count is taken from the low six bits of src1 regardless of the src1 type and treated as an unsigned integer in the range 0 to 63. Otherwise 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. The operation uses QWord mode if src0 or dst has the Q or UQ type but not if src1 is the only operand with the Q or UQ type.</p>			
Format: [(pred)] shr[.cmod] (exec_size) dst src0 src1			
Syntax			
[(pred)] shr[.cmod] (exec_size) reg reg reg [(pred)] shr[.cmod] (exec_size) reg reg imm32			
Pseudocode			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n++) { if (WrEn.chan[n]) { shiftCnt = src0 or dst has Q or UQ type ? src1.chan[n] & 0x3F : src1.chan[n] & 0x1F dst.chan[n] = src0.chan[n] » shiftCnt; } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types		Dst Types	
UB, UW, UD		UB, UW, UD	
UW, UD		UW, UD	
UW, UD		UQ	
UQ		UW, UD	
UQ		UQ	
DWord	Bit	Description	
0..3	127:64	RegSource Exists If: ([RegSource][Src1.RegFile] != 'IMM') Format: EU_INSTRUCTION_SOURCES_REG_REG	



shr - Shift Right		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
		Format: EU_INSTRUCTION_SOURCES_REG_IMM
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



SIMD8 Render Target Read MSD

MSD_RTR_SIMD8 - SIMD8 Render Target Read MSD		
Source:	Render Cache DataPort	
Length Bias:	1	
Family:	Other	
Group:	Render Target R/W	
DWord	Bit	Description
0	31	Reserved Format: MBZ Ignored
	30	Message Precision Subtype Default Value: 0h Format: Opcode Full precision data message <div style="background-color: #e6f2ff; text-align: center; padding: 2px;">Programming Notes</div> This field must be programmed to 0
	29	Reserved Format: MBZ Ignored
	28:25	Message Length Format: U4 Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.
	24:20	Response Length Format: U5 Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.
	19	Header Present Format: MDC_MHP If set, indicates that the message includes the 2-register header.



MSD_RTR_SIMD8 - SIMD8 Render Target Read MSD					
18	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
Format:	MBZ				
17:14	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>0Dh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Render Target Read message</p>	Default Value:	0Dh	Format:	Opcode
Default Value:	0Dh				
Format:	Opcode				
13	<p>Per-Sample PS Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, PS reads color phases on per sample basis for each slot.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE. When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_CO as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0. When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples). When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>	Format:	Enable	Programming Notes	
Format:	Enable				
Programming Notes					
12	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
Format:	MBZ				
11	<p>Slot Group Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>MDC_RT_SGS</td> </tr> </table> <p>This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.</p>	Format:	MDC_RT_SGS		
Format:	MDC_RT_SGS				
10:9	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
Format:	MBZ				



MSD_RTR_SIMD8 - SIMD8 Render Target Read MSD	
8	Render Target Message Subtype
	Default Value: 1h
	Format: Opcode
	SIMD8 single source message. Use slots [7:0] for the pixel enables, X/Y addresses, and oMask. Programming Notes The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [23:16] are referenced instead of [7:0].
7:0	Binding Table Index
	Format: MDC_BTS Specifies the Binding Table Index for the message



SIMD8 Render Target Write MSD

MSD_RTW_SIMD8 - SIMD8 Render Target Write MSD		
Source:	Render Cache DataPort	
Length Bias:	1	
Family:	Other	
Group:	Render Target R/W	
DWord	Bit	Description
0	31	Reserved
		Format: MBZ Ignored
	30	Message Precision Subtype
		Default Value: 0h
		Format: Opcode Full precision data message
	29	Reserved
		Format: MBZ Ignored
	28:25	Message Length
Format: U4 Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.		
24:20	Response Length	
	Format: U5 Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.	
19	Header Present	
	Format: MDC_MHP If set, indicates that the message includes the 2-register header.	
18	Reserved	
	Format: MBZ Ignored	



MSD_RTW_SIMD8 - SIMD8 Render Target Write MSD					
17:14	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0Ch</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Render Target Write message</p>	Default Value:	0Ch	Format:	Opcode
Default Value:	0Ch				
Format:	Opcode				
13	<p>Per-Sample PS Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.</p> <p style="text-align: center;">Programming Notes</p> <p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE.</p> <p>When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_CO as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0.</p> <p>When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples).</p> <p>When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>	Format:	Enable		
Format:	Enable				
12	<p>Last Render Target Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.</p> <p style="text-align: center;">Programming Notes</p> <p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set. In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>	Format:	Enable		
Format:	Enable				
11	<p>Slot Group Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MDC_RT_SGS</td> </tr> </table> <p>This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.</p>	Format:	MDC_RT_SGS		
Format:	MDC_RT_SGS				



MSD_RTW_SIMD8 - SIMD8 Render Target Write MSD	
10:8	Render Target Message Subtype
	Default Value: 4h
	Format: Opcode
	SIMD8 single source message. Use slots [7:0] for pixel enables, X/Y addresses, and oMask.
Programming Notes	
The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [23:16] are referenced instead of [7:0].	
7:0	Binding Table Index
	Format: MDC_BTS
Specifies the Binding Table Index for the message	



SIMD16 Render Target Read MSD

MSD_RTR_SIMD16 - SIMD16 Render Target Read MSD		
Source:	Render Cache DataPort	
Length Bias:	1	
Family:	Other	
Group:	Render Target R/W	
DWord	Bit	Description
0	31	Reserved
		Format: MBZ Ignored
	30	Message Precision Subtype
		Default Value: 0h
		Format: Opcode
		Full precision data message Programming Notes This field must be programmed to 0
29	Reserved	
	Format: MBZ Ignored	
28:25	Message Length	
	Format: U4 Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.	
24:20	Response Length	
	Format: U5 Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.	
19	Header Present	
	Format: MDC_MHP If set, indicates that the message includes the 2-register header.	



MSD_RTR_SIMD16 - SIMD16 Render Target Read MSD					
18	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
Format:	MBZ				
17:14	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>0Dh</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Render Target Read message</p>	Default Value:	0Dh	Format:	Opcode
Default Value:	0Dh				
Format:	Opcode				
13	<p>Per-Sample PS Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>If set, PS reads color phases on per sample basis for each slot.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE. When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_CO as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0. When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples). When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>	Format:	Enable	Programming Notes	
Format:	Enable				
Programming Notes					
12	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
Format:	MBZ				
11	<p>Slot Group Select</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>MDC_RT_SGS</td> </tr> </table> <p>This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.</p>	Format:	MDC_RT_SGS		
Format:	MDC_RT_SGS				
10:9	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ		
Format:	MBZ				



MSD_RTR_SIMD16 - SIMD16 Render Target Read MSD	
8	Render Target Message Subtype
	Default Value: 0h
	Format: Opcode
	SIMD16 single source message. Use slots [15:0] for the pixel enables, X/Y addresses, and oMask.
Programming Notes	
The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [31:16] are referenced instead of [15:0].	
7:0	Binding Table Index
	Format: MDC_BT
Specifies the Binding Table Index for the message	



SIMD16 Render Target Write MSD

MSD_RTW_SIMD16 - SIMD16 Render Target Write MSD		
Source:	Render Cache DataPort	
Length Bias:	1	
Family:	Other	
Group:	Render Target R/W	
DWord	Bit	Description
0	31	Reserved
		Format: MBZ Ignored
	30	Message Precision Subtype
		Default Value: 0h
		Format: Opcode Full precision data message
	29	Reserved
		Format: MBZ Ignored
	28:25	Message Length
Format: U4 Specifies the number of 256-bit GRF registers sent as the message payload (including the header). Valid value ranges are 1 to 15.		
24:20	Response Length	
	Format: U5 Specifies the number of 256-bit GRF registers expected as the message response payload. Valid value ranges are 0 to 16.	
19	Header Present	
	Format: MDC_MHP If set, indicates that the message includes the 2-register header.	
18	Reserved	
	Format: MBZ Ignored	

**MSD_RTW_SIMD16 - SIMD16 Render Target Write MSD**

17:14	Message Type	
	Default Value:	0Ch
	Format:	Opcode
Render Target Write message		
13	Per-Sample PS Enable	
	Format:	Enable
	If set, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot.	
Programming Notes		
<p>This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE.</p> <p>When this bit is set and PS runs at pixel-frequency, i.e. pixel shader dispatch mode is PER_PIXEL, Render Target read and write messages interpret bits 9:6 in MCH_RT_CO as sample index. In this mode, render target write message payload and render target read writeback payload contain color of a specific sample in all dispatched pixels. RT writes referring to out-of-bound samples have no effect. RT reads from out-of-bound samples return 0.</p> <p>When this bit is clear and PS runs at pixel-frequency, render target write messages contain color value for entire pixel (all samples).</p> <p>When this bit is clear and PS runs at pixel-frequency, render target reads are disallowed per API spec (RT read without specifying sample index forces sample-frequency dispatch). HW behavior is undefined in such case.</p>		
12	Last Render Target Select	
	Format:	Enable
	This bit must be set on the last render target write message sent for each group of pixels. For single render target pixel shaders, this bit is set on all render target write messages. For multiple render target pixel shaders, this bit is set only on messages sent to the last render target. This bit must be zero for SIMD8 Image Write message. In general, when threads are not launched by 3D FF, this bit must be zero.	
Programming Notes		
<p>When a pixel shader has render target writes at finer granularity than the dispatch rate, last render target write to a null surface must be present at the dispatch rate with this bit set.</p> <p>In particular, if a kernel is dispatched at pixel rate and it only writes to render targets at sample-rate, it must include a pixel-rate render target write to a null surface with Last Render Target Select bit enabled.</p>		
11	Slot Group Select	
	Format:	MDC_RT_SGS
This field selects whether slots 15:0 or slots 31:16 are used for bypassed data.		



MSD_RTW_SIMD16 - SIMD16 Render Target Write MSD						
10:8	<p>Render Target Message Subtype</p> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>SIMD16 Single source message. Use slots [15:0] for pixel enables, X/Y addresses, and oMask.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>The above slots indicated are within the 16 slots selected by Slot Group Select. If SLOTGRP_HI is selected, slots [31:16] are referenced instead of [15:0].</p>	Default Value:	0h	Format:	Opcode	Programming Notes
Default Value:	0h					
Format:	Opcode					
Programming Notes						
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MDC_BTS</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS			
Format:	MDC_BTS					



Split Send Message

sends - Split Send Message	
Source:	Eulsa
Length Bias:	4
Description	
<p>The sends 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 sends 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. <src0> and <src1> are the lead GRF registers for the first and second block of the request respectively. <dest> is the lead GRF register for response. The message descriptor field <desc> contains the Message Length (the number of consecutive GRF registers corresponding to src0) and the Response Length (the number of consecutive GRF registers). It also contains the header present bit, and the function control signals. The extend message descriptor field <ex_desc> contains the target function ID, the Extended Message Length (the number of consecutive GRF registers corresponding to src1) and the extended function control signals. WrEn is forwarded to the target function in the message sideband.</p> <p>The sends 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.</p> <p>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 31 bits of <reg32a> are used.</p> <p>Extended Message descriptor field <ex_desc> can be a 32-bit immediate, imm32 or a 32bit scalar register, <reg32a>. The bits3:0 of the <ex_desc> specifies the SFID for the message. The EOT field always comes from bit127 of the instruction word, which is the bit5 of <ex_desc>. A thread must terminate with a sends instruction with EOT turned on. The bits9:6 of <ex_desc> specify the extended message length and bits31:12 specify the 20bit extended function control. Interpretation of the extended function control signals is subject to the target external function. The scalar register <reg32a> is selected when SelReg32ExDesc is set, ExDesc.RegNum[3:0] provides the addressing for reg32a for extended message descriptor. This selects one of the index sub registers. Subregisters selected are always aligned to dword. This implies, the even index subregisters must be used. </reg32a> </ex_desc> </ex_desc> </ex_desc> </reg32a> </ex_desc></p> <p>Function control is now extended to 20 bits as specified in the below definition.</p> <p><src0> is a 256-bit aligned GRF register. It serves as the leading GRF register of the request.</p> <p><src1> is a 256-bit aligned GRF register or a null register. It serves as the leading GRF register for the second block of the request when it is not a null register. It is required that the second block of GRFs does not overlap with the first block. If it is a null register the Extended Message Length must be 0. The sum of Message Length and Extended Message Length must not be greater than 15.</p> <p>The source dependency control, {NoSrcDepSet} is used to control the setting of source dependency for both the sources.</p> <p><dest> serves for two purposes: to provide the leading GRF register location for the response message if</p>	



sends - Split Send Message

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 destination type field is always valid and is used to generate 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 address immediates for indirect sources and destination must be oword aligned.
 The 16-bit channel enables of the message sideband are formed based on the WrEn. Interpretation of the channel enable sideband signals is subject to the target external function. In general for a 'sends' 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.
 NoDDClr and NoDDChk must not be used for send instruction.

Send a message stored in GRF locations starting at <src0> followed by <src1> to a shared function identified by <ex_desc> along with control from <desc> and <ex_desc> with a GRF writeback location at <dest>.

Format:

[(pred)] sends (exec_size) <dest> <src0> <src1> <ex_desc> <desc>

Restriction

Restriction : Software must obey the following rules in signaling the end of thread using the sends instruction: The posted destination operand must be null. No acknowledgement is allowed for the sends 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 sends instruction with message to a shared function on the output message bus; therefore, it cannot terminate with a sends 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 sends instruction with message to the Thread Spawner unit. A child thread should also terminate with a sends to TS. Please refer to the Media Chapter for more detailed description. The sends instruction can not update accumulator registers. Saturate is not supported for sends instruction. ThreadCtrl encodings Switch is not supported for sends instruction. The sends with EOT should use register space R112-R127 for <src>. This is to enable loading of a new thread into the same slot while the message with EOT for current thread is pending dispatch. Any instruction updating the ARF must use a {Switch} if the ARF is not used before EOT.</src>

Syntax

```
[(pred)] sends (exec_size) reg greg greg imm32 imm32 [(pred)] sends (exec_size)
reg greg greg imm32 reg32a [(pred)] sends (exec_size) reg greg reg reg32a imm32
[(pred)] sends (exec_size) reg greg reg reg32a reg32a
```



sends - Split Send Message

Pseudocode

```
Evaluate(WrEn); <MsgChEnable> = WrEn; <SourceReg0> = <src0>.RegNum; <SourceReg1>
= <src1>.RegNum; MessageEnqueue(<MsgChEnable>, <ResponseReg>, <SourceReg0>,
<SourceReg1>, <ex_desc>, <dest>);
```

Predication	Conditional Modifier	Saturation	Source Modifier
Y	N	N	N

DWord	Bit	Description
0..3	127:96	Message Format: EU_INSTRUCTION_OPERAND_SEND_MSG
	95:80	ExDesc[31:16] Format: ExtMsgDescpt[31:16]
	79	Source 0 Addressing Mode Format: AddrMode
	78	Reserved Exists If: ([Source 0 Addressing Mode]='Direct') Format: MBZ
	78	Source 0 Address Immediate Sign [9] Exists If: ([Source 0 Addressing Mode]='Indirect') Format: S9[9]
	77	SelReg32Desc
	76:73	Source 0 Address Subregister Number Exists If: ([Source 0 Addressing Mode]='Indirect')
	76:69	Source 0 Register Number Exists If: ([Source 0 Addressing Mode]='Direct')
	72:68	Source 0 Address Immediate [8:4] Exists If: ([Source 0 Addressing Mode]='Indirect') Format: S9[8:4]
	68	Source 0 Subregister Number Exists If: ([Source 0 Addressing Mode]='Direct')
	67:64	ExDesc[9:6] Format: ExtMsgDescpt[9:6]
	63	Destination Addressing Mode Format: AddrMode



sends - Split Send Message		
62	Destination Address Immediate Sign [9]	
	Exists If:	((Destination Addressing Mode) == 'Indirect')
	Format:	S9[9]
62	Reserved	
	Exists If:	((Destination Addressing Mode) == 'Direct')
	Format:	MBZ
61	Reserved	
	Format:	MBZ
60:57	Destination Address Subregister Number	
	Exists If:	((Destination Addressing Mode) == 'Indirect')
60:53	Destination Register Number	
	Exists If:	((Destination Addressing Mode) == 'Direct')
56:52	Destination Address Immediate [8:4]	
	Exists If:	((Destination Addressing Mode) == 'Indirect')
	Format:	S9[8:4]
52	Destination Subregister Number [4]	
	Exists If:	((Destination Addressing Mode) == 'Direct')
51:44	Source 1 Register Number	
43:41	Reserved	
	Format:	MBZ
40:37	Destination Type	
36	Source 1 Register File	
	Format:	RegFile[0]
35	Destination Register File	
	Format:	RegFile[0]
34	MaskCtrl	
33:32	Flag Register Number/Subregister Number	
31:28	Controls B	
	Format:	EU_INSTRUCTION_CONTROLS_B
27:24	Shared Function ID (SFID)	
	Format:	SFID
23:8	Controls A	
	Format:	EU_INSTRUCTION_CONTROLS_A
7	Reserved	
	Format:	MBZ



sends - Split Send Message		
	6:0	Opcode
		Format: EU_OPCODE



STATE_BASE_ADDRESS

STATE_BASE_ADDRESS								
Source:	BSpec							
Length Bias:	2							
<p>The STATE_BASE_ADDRESS command sets the base pointers for subsequent state, instruction, and media indirect object accesses by the GPE.</p> <p>For more information see the Base Address Utilization table in the Memory Access Indirection narrative topic.</p>								
Programming Notes								
<p>The following commands must be reissued following any change to the base addresses:</p> <ul style="list-style-type: none"> • 3DSTATE_CC_POINTERS • 3DSTATE_BINDING_TABLE_POINTERS • 3DSTATE_SAMPLER_STATE_POINTERS • 3DSTATE_VIEWPORT_STATE_POINTERS • MEDIA_STATE_POINTERS <p>Execution of this command causes a full pipeline flush, thus its use should be minimized for higher performance.</p> <p>SW must always program PIPE_CONTROL with "CS Stall" and "Render Target Cache Flush Enable" set before programming STATE_BASE_ADDRESS command for GPGPU workloads i.e when pipeline select is GPGPU via PIPELINE_SELECT command. This is required to achieve better GPGPU preemption latencies in certain workload programming sequences. If programming PIPE_CONTROL has performance implications then preemption latencies can be traded off against performance by not implementing this programming note.</p>								
DWord	Bit	Description						
0	31:29	Command Type Default Value: 3h GFXPIPE						
	28:27	Command SubType Default Value: 0h GFXPIPE_COMMON						
	26:24	3D Command Opcode Default Value: 1h GFXPIPE_NONPIPELINED						
	23:16	3D Command Sub Opcode Default Value: 01h STATE_BASE_ADDRESS						
	15:8	Reserved Format: MBZ						
	7:0	DWord Length Format: =n Total Length - 2						
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	Value	Name	Description			
Value	Name	Description						



STATE_BASE_ADDRESS			
	11h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)
	10h	DWORD_COUNT_modify	Excludes DWord (0,1) To avoid updating the "Bindless Surface State Base Size" when the "Bindless Surface State Base Address Modify Enable" bit is cleared then the size of the command must be reduced to one less than the size of the instruction.
1..2	63:12	General State Base Address	
		Format:	GraphicsAddress[63:12]
		Specifies the 4K-byte aligned base address for general state accesses. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].	
		Programming Notes	
		Bounds checking is performed on general state accesses by Data Port Shared Functions for stateless A32 messages. Bounds checking is enabled when General State Base Address [46:12] + General State Buffer Size [31:12] is $\leq 2^{47}$. This ensures that the General State Buffer does not straddle the canonical address boundary where GraphicsAddress [47] changes.	
		Restriction	
		Restriction : General State Base Address [47:12] + General State Buffer Size [31:12] must be $< 2^{48}$. It is illegal programming for this to be $\geq 2^{48}$.	
		Restriction : When using stateless (A32) Data Port messages, General State Base Address [47:12] + Buffer Base Address [31:0] must be $< 2^{48}$. It is illegal for this to be $\geq 2^{48}$.	
	11	Reserved	
		Format:	MBZ
	10:4	General State Memory Object Control State	
		Format:	MEMORY_OBJECT_CONTROL_STATE
		Specifies the memory object control state for indirect state using the General State Base Address , with the exception of the stateless data port accesses.	
	3:1	Reserved	
		Format:	MBZ
	0	General State Base Address Modify Enable	
		Format:	Enable
		The other fields in this DWord and the following DWord are updated only when this bit is set.	
		Value	Name
		Description	
		0h	Disable
		1h	Enable
		Ignore the updated address.	Modify the address.



STATE_BASE_ADDRESS												
3	31:23	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	22:16	<p>Stateless Data Port Access Memory Object Control State</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for stateless data port accesses.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE								
Format:	MEMORY_OBJECT_CONTROL_STATE											
15:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
4..5	63:12	<p>Surface State Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[63:12]</td> </tr> </table> <p>Specifies the 4K-byte aligned base address for binding table and surface state accesses. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p>	Format:	GraphicsAddress[63:12]								
	Format:	GraphicsAddress[63:12]										
	11	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	10:4	<p>Surface State Memory Object Control State</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for indirect state using the Surface State Base Address.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE								
Format:	MEMORY_OBJECT_CONTROL_STATE											
3:1	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
0	<p>Surface State Base Address Modify Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>The other fields in this DWord and the following DWord are updated only when this bit is set.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Ignore the updated address.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Modify the address.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Setting this bit to 1 in a batch buffer causes the resource streamer to stop; for performance reasons the SW should only place commands with this bit set in the ring buffer.</p> <p>Before programming the Surface State Base Address, the RS must be disabled. Within a batch buffer where the RS is enabled, RS may be disabled thru a MI_RS_CONTROL command with Resource Streamer Control cleared prior to the STATE_BASE_ADDRESS with Surface State Base Address Modify Enable set and then re-enabled with another MI_RS_CONTROL with Resource Streamer Control set.</p>	Format:	Enable	Value	Name	Description	0h	Disable	Ignore the updated address.	1h	Enable	Modify the address.
Format:	Enable											
Value	Name	Description										
0h	Disable	Ignore the updated address.										
1h	Enable	Modify the address.										



STATE_BASE_ADDRESS									
6..7	63:12 Dynamic State Base Address Format: GraphicsAddress[63:12] Specifies the 4K-byte aligned base address for sampler and viewport state accesses. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].								
	11 Reserved Format: MBZ								
	10:4 Dynamic State Memory Object Control State Format: MEMORY_OBJECT_CONTROL_STATE Specifies the memory object control state for indirect state using the Dynamic State Base Address . Push constants defined in 3DSTATE_CONSTANT_(VS GS PS) commands do not use this control state, although they can use the corresponding base address. The memory object control state for push constants is defined within the command.								
	3:1 Reserved Format: MBZ								
	0 Dynamic State Base Address Modify Enable Format: Enable The other fields in this DWord and the following DWord are updated only when this bit is set. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Ignore the updated address.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Modify the address.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	Ignore the updated address.	1h	Enable
Value	Name	Description							
0h	Disable	Ignore the updated address.							
1h	Enable	Modify the address.							
8..9	63:12 Indirect Object Base Address Format: GraphicsAddress[63:12]IndirectObject Specifies the 4K-byte aligned base address for indirect object load in MEDIA_OBJECT command.								
	11 Reserved Format: MBZ								
	10:4 Indirect Object Memory Object Control State Format: MEMORY_OBJECT_CONTROL_STATE Specifies the memory object control state for indirect objects using the Indirect Object Base Address .								
	3:1 Reserved Format: MBZ								
	0 Indirect Object Base Address Modify Enable Format: Enable The other fields in this DWord and the following DWord are updated only when this bit is set.								



STATE_BASE_ADDRESS												
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%; text-align: center;">Value</th> <th style="width: 20%; text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">Disable</td> <td>Ignore the updated address.</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">Enable</td> <td>Modify the address.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable	Ignore the updated address.	1h	Enable	Modify the address.	
Value	Name	Description										
0h	Disable	Ignore the updated address.										
1h	Enable	Modify the address.										
10..11	63:12	Instruction Base Address <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[63:12]</td> </tr> </table> <p>Specifies the 4K-byte aligned base address for all EU instruction accesses. GraphicsAddress[63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p>	Format:	GraphicsAddress[63:12]								
	Format:	GraphicsAddress[63:12]										
	11	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	10:4	Instruction Memory Object Control State <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td style="color: red;">MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for EU instructions using the Instruction Base Address.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE								
Format:	MEMORY_OBJECT_CONTROL_STATE											
3:1	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
0	Instruction Base Address Modify Enable <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>The other fields in this DWord and the following DWord are updated only when this bit is set.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%; text-align: center;">Value</th> <th style="width: 20%; text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">Disable</td> <td>Ignore the updated address.</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">Enable</td> <td>Modify the address.</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0h	Disable	Ignore the updated address.	1h	Enable	Modify the address.
Format:	Enable											
Value	Name	Description										
0h	Disable	Ignore the updated address.										
1h	Enable	Modify the address.										
12	31:12	General State Buffer Size <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>U20</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 100%;">FormatDesc</td> </tr> </table> <p>This field specifies the size of the buffer in 4K pages. Any access that straddles or goes past the end of the buffer returns 0. Note that BufferSize=0 indicates that there is no valid data in the buffer.</p>	Format:	U20	FormatDesc							
	Format:	U20										
	FormatDesc											
11:1	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
0	General State Buffer Size Modify Enable <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>The bound in this DWord is updated only when this bit is set.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%; text-align: center;">Value</th> <th style="width: 20%; text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">Disable</td> <td>Ignore the updated bound.</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0h	Disable	Ignore the updated bound.			
Format:	Enable											
Value	Name	Description										
0h	Disable	Ignore the updated bound.										



STATE_BASE_ADDRESS																	
		<table border="1"> <tr> <td>1h</td> <td>Enable</td> <td>Modify the updated bound.</td> </tr> </table>	1h	Enable	Modify the updated bound.												
1h	Enable	Modify the updated bound.															
13	31:12	Dynamic State Buffer Size <table border="1"> <tr> <td>Format:</td> <td>U20</td> </tr> <tr> <td colspan="2">FormatDesc</td> </tr> <tr> <td colspan="2">This field specifies the size of the buffer in 4K pages. Any access that straddles or goes past the end of the buffer returns 0. Note that BufferSize=0 indicates that there is no valid data in the buffer.</td> </tr> </table>	Format:	U20	FormatDesc		This field specifies the size of the buffer in 4K pages. Any access that straddles or goes past the end of the buffer returns 0. Note that BufferSize=0 indicates that there is no valid data in the buffer.										
		Format:	U20														
		FormatDesc															
		This field specifies the size of the buffer in 4K pages. Any access that straddles or goes past the end of the buffer returns 0. Note that BufferSize=0 indicates that there is no valid data in the buffer.															
	11:1	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	0	Dynamic State Buffer Size Modify Enable <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> <tr> <td colspan="2">FormatDesc</td> </tr> <tr> <td colspan="2">The bound in this DWord is updated only when this bit is set.</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Ignore the updated bound.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Modify the updated bound.</td> </tr> </tbody> </table>	Format:	Enable	FormatDesc		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 updated bound.
		Format:	Enable														
		FormatDesc															
		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 updated bound.															
14	31:12	Indirect Object Buffer Size <table border="1"> <tr> <td>Format:</td> <td>U20</td> </tr> <tr> <td colspan="2">FormatDesc</td> </tr> <tr> <td colspan="2">This field specifies the size of the buffer in 4K pages. Any access that straddles or goes past the end of the buffer returns 0. Note that BufferSize=0 indicates that there is no valid data in the buffer.</td> </tr> </table>	Format:	U20	FormatDesc		This field specifies the size of the buffer in 4K pages. Any access that straddles or goes past the end of the buffer returns 0. Note that BufferSize=0 indicates that there is no valid data in the buffer.										
		Format:	U20														
		FormatDesc															
		This field specifies the size of the buffer in 4K pages. Any access that straddles or goes past the end of the buffer returns 0. Note that BufferSize=0 indicates that there is no valid data in the buffer.															
	11:1	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
	Format:	MBZ															
	0	Indirect Object Buffer Size Modify Enable <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> <tr> <td colspan="2">FormatDesc</td> </tr> <tr> <td colspan="2">The bound in this DWord is updated only when this bit is set.</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Ignore the updated bound.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Modify the updated bound.</td> </tr> </tbody> </table>	Format:	Enable	FormatDesc		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 updated bound.
		Format:	Enable														
		FormatDesc															
		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 updated bound.															



STATE_BASE_ADDRESS									
15	31:12	<p>Instruction Buffer Size</p> <table border="1"> <tr> <td>Format:</td> <td>U20</td> </tr> </table> <p>FormatDesc</p> <p>This field specifies the size of the buffer in 4K pages. Any access that straddles or goes past the end of the buffer returns 0. Note that BufferSize=0 indicates that there is no valid data in the buffer.</p>	Format:	U20					
	Format:	U20							
	11:1	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
0	<p>Instruction Buffer size Modify Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>FormatDesc</p> <p>The bound in this DWord is updated only when this bit is set.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Ignore the updated bound.</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0h	Disable	Ignore the updated bound.
Format:	Enable								
Value	Name	Description							
0h	Disable	Ignore the updated bound.							
63:12	<p>Bindless Surface State Base Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[63:12]</td> </tr> </table> <p>Specifies the 4K-byte aligned base address for bindless surface state accesses. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p>	Format:	GraphicsAddress[63:12]						
Format:	GraphicsAddress[63:12]								
16..17	11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
	10:4	<p>Bindless Surface State Memory Object Control State</p> <table border="1"> <tr> <td>Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>Specifies the memory object control state for indirect state using the Bindless Surface State Base Address.</p>	Format:	MEMORY_OBJECT_CONTROL_STATE					
	Format:	MEMORY_OBJECT_CONTROL_STATE							
	3:1	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ								
0	<p>Bindless Surface State Base Address Modify Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p style="text-align: center;">Description</p> <p>The other fields in this DWord are updated only when this bit is set.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> </tbody> </table>	Format:	Enable	Value	Name	Description			
Format:	Enable								
Value	Name	Description							



STATE_BASE_ADDRESS								
		<table border="1"><tr><td>0h</td><td>Disable</td><td>Ignore the updated address</td></tr><tr><td>1h</td><td>Enable</td><td>Modify the address</td></tr></table>	0h	Disable	Ignore the updated address	1h	Enable	Modify the address
0h	Disable	Ignore the updated address						
1h	Enable	Modify the address						
18	31:12	<p>Bindless Surface State Size</p> <table border="1"><tr><td>Format:</td><td>U20</td></tr></table> <p>This field indicates the size-1 of the Bindless Surface State buffer in 64-Byte increments. Any SSO beyond this maximum size points to offset 0. Example: If the buffer contains 512 surface states, then this field must be programmed to 0x1FF (511 decimal).</p>	Format:	U20				
Format:	U20							
	11:0	<p>Reserved</p> <table border="1"><tr><td>Format:</td><td>MBZ</td></tr></table>	Format:	MBZ				
Format:	MBZ							



STATE_PREFETCH

STATE_PREFETCH						
Source:	BSpec					
Length Bias:	2					
<p>(This command is provided strictly for performance optimization opportunities, and likely requires some experimentation to evaluate the overall impact of additional prefetching.)</p> <p>The STATE_PREFETCH command causes the GPE to attempt to prefetch a sequence of 64-byte cache lines into the GPE-internal cache ("L2 ISC") used to access EU kernel instructions and fixed/shared function indirect state data. While state descriptors, surface state, and sampler state are automatically prefetched by the GPE, this command may be used to prefetch data not automatically prefetched, such as: 3D viewport state; Media pipeline Interface Descriptors; EU kernel instructions.</p>						
Restriction						
Due to know HW issue this command doesn't achieve its intended purpose and must not be exercised/programmed by SW.						
DWord	Bit	Description				
0	31:29	Command Type Default Value: 3h GFXPIPE				
	28:27	Command SubType Default Value: 0h GFXPIPE_COMMON				
	26:24	3D Command Opcode Default Value: 0h GFXPIPE_PIPELINED				
	23:16	3D Command Sub Opcode Default Value: 03h STATE_PREFETCH				
	15:8	Reserved Format: MBZ				
	7:0	DWord Length Format: =n Total Length - 2				
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>DWORD_COUNT_n [Default]</td> <td>Excludes DWord (0,1)</td> </tr> </tbody> </table>	Value	Name	Description	0h
Value	Name	Description				
0h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)				
1	31:6	Prefetch Pointer Format: GraphicsAddress[31:6] Specifies the 64-byte aligned address to start the prefetch from. This pointer is an absolute virtual address, it is not relative to any base pointer.				
	5:3	Reserved Format: MBZ				



STATE_PREFETCH								
	2:0	Prefetch Count						
		Format: U3-1 count of cache lines						
		Indicates the number of contiguous 64-byte cache lines that will be prefetched.						
		<table border="1"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>[0,7]</td><td></td><td>indicating a count of [1,8]</td></tr></tbody></table>	Value	Name	Description	[0,7]		indicating a count of [1,8]
		Value	Name	Description				
[0,7]		indicating a count of [1,8]						



STATE_SIP

STATE_SIP								
Source:	BSpec							
Length Bias:	2							
The STATE_SIP command specifies the starting instruction location of the System Routine that is shared by all threads in execution.								
DWord	Bit	Description						
0	31:29	Command Type Default Value: 3h GFXPIPE						
	28:27	Command SubType Default Value: 0h GFXPIPE_COMMON						
	26:24	3D Command Opcode Default Value: 1h GFXPIPE_NONPIPELINED						
	23:16	3D Command Sub Opcode Default Value: 02h STATE_SIP						
	15:8	Reserved Format: MBZ						
	7:0	DWord Length Format: =n Total Length - 2 <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td>DWORD_COUNT_n [Default]</td> <td>Excludes DWord (0,1)</td> </tr> </tbody> </table>	Value	Name	Description	1h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)
Value	Name	Description						
1h	DWORD_COUNT_n [Default]	Excludes DWord (0,1)						
1..2	63:4	System Instruction Pointer Format: InstructionBaseOffset[63:4]Kernel Specifies the instruction address of the system routine associated with the current context 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. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47]. <table border="1"> <thead> <tr> <th colspan="3">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="3">This portion of the command is not context save/restored. The context image may restore this command as a 2 dword command rather than a 3 dword command.</td> </tr> </tbody> </table>	Programming Notes			This portion of the command is not context save/restored. The context image may restore this command as a 2 dword command rather than a 3 dword command.		
	Programming Notes							
This portion of the command is not context save/restored. The context image may restore this command as a 2 dword command rather than a 3 dword command.								
3:0	Reserved Format: MBZ							



Sum of Absolute Difference 2

sad2 - Sum of Absolute Difference 2			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>The results are also stored in the accumulator register. The destination operand and the accumulator maintain 16 bits per channel precision.</p> <p>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.</p>			
<p>Format:</p> <p>[(pred)] sad2[.cmod] (exec_size) dst src0 src1</p>			
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			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n += 2) { if (WrEn.chan[n]) { dst.chan[n] = abs(src0.chan[n] - src1.chan[n]) + abs(src0.chan[n+1] - src1.chan[n+1]); } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
B, UB	W, UW		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG



sad2 - Sum of Absolute Difference 2		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Sum of Absolute Difference Accumulate 2

sada2 - Sum of Absolute Difference Accumulate 2			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>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.</p> <p>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.</p>			
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			
<pre>Evaluate(WrEn); for (n = 0; n < exec_size; n += 2) { uwTmp = abs(src0.chan[n] - src1.chan[n]) + abs(src0.chan[n+1] - src1.chan[n+1]); if (WrEn.chan[n]) { dst.chan[n] = uwTmp + acc[n]; } }</pre>			
Predication	Conditional Modifier	Saturation	Source Modifier
Y	Y	Y	Y
Src Types	Dst Types		
B, UB	W, UW		
DWord	Bit	Description	
0..3	127:64	RegSource	
		Exists If:	([RegSource][Src1.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_SOURCES_REG_REG



sada2 - Sum of Absolute Difference Accumulate 2		
	127:64	ImmSource
		Exists If: ([ImmSource][Src1.RegFile] = 'IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:32	Operand Controls
		Format: EU_INSTRUCTION_OPERAND_CONTROLS
	31:0	Header
Format: EU_INSTRUCTION_HEADER		



Typed Surface Read MSD

MSD1R_TS - Typed Surface Read MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Surface R/W	
Group:	Scattered Typed Surface R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type
		Default Value: 05h
		Format: Opcode Typed Surface Read message
13:12	Slot Group	
Format: MDC_SG3 Specifies the Slot Group mode of the message (which slots are processed)		
11:8	Channel Mask	
Format: MDC_CMASK Specifies which RGBA channels are included in the message payload.		
7:0	Binding Table Index	
Format: MDC_BTS Specifies the Binding Table Index for the message		



Typed Surface Write MSD

MSD1W_TS - Typed Surface Write MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Typed Surface R/W	
Group:	Scattered Typed Surface R/W	
DWord	Bit	Description
0	19	Header Present
		Format: MDC_MHP If set, indicates that the message includes the header.
	18:14	Message Type
		Default Value: 0Dh
		Format: Opcode Typed Surface Write message
13:12	Slot Group	
	Format: MDC_SG3 Specifies the Slot Group mode of the message (which slots are processed)	
11:8	Channel Mask	
	Format: MDC_CMASK Specifies which RGBA channels are included in the message payload.	
7:0	Binding Table Index	
	Format: MDC_BTS Specifies the Binding Table Index for the message	



Untyped Surface Read MSD

MSD1R_US - Untyped Surface Read MSD		
Source:	DataPort 1	
Length Bias:	1	
Family:	Untyped Surface R/W	
Group:	Scattered Untyped Surface R/W	
DWord	Bit	Description
0	19	Header Present Format: MDC_MHP If set, indicates that the message includes the header.
		Message Type Default Value: 01h Format: Opcode Untyped Surface Read message
	13:12	SIMD Mode Format: MDC_SM3 Specifies the SIMD mode of the message (number of slots processed)
		Channel Mask Format: MDC_CMASK Specifies which RGBA channels are included in the message payload.
	7:0	Binding Table Index Format: MDC_BTS_SLM_A32 Specifies the Binding Table Index for the message



Untyped Surface Write MSD

MSD1W_US - Untyped Surface Write MSD						
Source:	DataPort 1					
Length Bias:	1					
Family:	Untyped Surface R/W					
Group:	Scattered Untyped Surface R/W					
DWord	Bit	Description				
0	19	<p>Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_MHP</td> </tr> </table> <p>If set, indicates that the message includes the header.</p>	Format:	MDC_MHP		
	Format:	MDC_MHP				
	18:14	<p>Message Type</p> <table border="1"> <tr> <td>Default Value:</td> <td>09h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table> <p>Untyped Surface Write message</p>	Default Value:	09h	Format:	Opcode
	Default Value:	09h				
	Format:	Opcode				
13:12	<p>SIMD Mode</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_SM3</td> </tr> </table> <p>Specifies the SIMD mode of the message (number of slots processed)</p>	Format:	MDC_SM3			
Format:	MDC_SM3					
11:8	<p>Channel Mask</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_UW_CMASK</td> </tr> </table> <p>Specifies which RGBA channels are included in the message payload.</p>	Format:	MDC_UW_CMASK			
Format:	MDC_UW_CMASK					
7:0	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td>MDC_BTS_SLM_A32</td> </tr> </table> <p>Specifies the Binding Table Index for the message</p>	Format:	MDC_BTS_SLM_A32			
Format:	MDC_BTS_SLM_A32					



URB Hword Dual Block Read MSD

MSD_UR_HWDB - URB Hword Dual Block Read MSD								
Source:		Read-Only DataPort						
Length Bias:		1						
DWord	Bit	Description						
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header.						
	18	Reserved Format: MBZ Ignored						
	17	Per Slot Offset Format: MHC_PSOP Specifies if per-slot offsets are present and will be added to the Global Offset .						
	16	Reserved Format: MBZ Ignored						
	15	Swizzle Control Format: Opcode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>URB_INTERLEAVED [Default]</td> <td>Use two URB entries (URB Handle 0 and URB Handle 1).</td> </tr> </tbody> </table>	Value	Name	Description	1	URB_INTERLEAVED [Default]	Use two URB entries (URB Handle 0 and URB Handle 1).
	Value	Name	Description					
	1	URB_INTERLEAVED [Default]	Use two URB entries (URB Handle 0 and URB Handle 1).					
	14:4	Global Offset Format: U11 Specifies the offset, in units of Hword elements, from the start of the URB handle for the access. If Per Slot Offset is set, the global offset is added to those offsets to form the overall offset. Range [0,1023]						
3:0	URB Opcode Format: Opcode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>URB_READ_HWORD [Default]</td> <td>URB Hword Read message</td> </tr> </tbody> </table>	Value	Name	Description	2	URB_READ_HWORD [Default]	URB Hword Read message	
Value	Name	Description						
2	URB_READ_HWORD [Default]	URB Hword Read message						



URB Hword Dual Block Write MSD

MSD_UW_HWDB - URB Hword Dual Block Write MSD								
Source:		Read-Only DataPort						
Length Bias:		1						
DWord	Bit	Description						
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header.						
	18	Reserved Format: MBZ Ignored						
	17	Per Slot Offset Format: MHC_PSOP Specifies if per-slot offsets are present and will be added to the Global Offset .						
	16	Reserved Format: MBZ Ignored						
	15	Swizzle Control Format: Opcode <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>URB_INTERLEAVED [Default]</td> <td>Use two URB entries (URB Handle 0 and URB Handle 1).</td> </tr> </tbody> </table>	Value	Name	Description	1	URB_INTERLEAVED [Default]	Use two URB entries (URB Handle 0 and URB Handle 1).
	Value	Name	Description					
	1	URB_INTERLEAVED [Default]	Use two URB entries (URB Handle 0 and URB Handle 1).					
	14:4	Global Offset Format: U11 Specifies the offset, in units of Hword elements, from the start of the URB handle for the access. If Per Slot Offset is set, the global offset is added to those offsets to form the overall offset. Range [0,1023]						
3:0	URB Opcode Format: Opcode <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>URB_WRITE_HWORD [Default]</td> <td>URB Hword Read message</td> </tr> </tbody> </table>	Value	Name	Description	0	URB_WRITE_HWORD [Default]	URB Hword Read message	
Value	Name	Description						
0	URB_WRITE_HWORD [Default]	URB Hword Read message						



URB Oword Block Write MSD

MSD_UW_OWB - URB Oword Block Write MSD								
Source:		Read-Only DataPort						
Length Bias:		1						
DWord	Bit	Description						
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header.						
	18	Reserved Format: MBZ Ignored						
	17	Per Slot Offset Format: MHC_PSOP Specifies if per-slot offsets are present and will be added to the Global Offset .						
	16	Reserved Format: MBZ Ignored						
	15	Swizzle Control Format: Opcode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>URB_NOSWIZZLE [Default]</td> <td>Use a single URB entry (URB Handle 0).</td> </tr> </tbody> </table>	Value	Name	Description	0	URB_NOSWIZZLE [Default]	Use a single URB entry (URB Handle 0).
	Value	Name	Description					
	0	URB_NOSWIZZLE [Default]	Use a single URB entry (URB Handle 0).					
	14:4	Global Offset Format: U11 Specifies the offset, in units of Oword elements, from the start of the URB handle for the access. If Per Slot Offset is set, the global offset is added to those offsets to form the overall offset. Range [0,2047]						
3:0	URB Opcode Format: Opcode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>URB_WRITE_OWORD [Default]</td> <td>URB Oword Write message</td> </tr> </tbody> </table>	Value	Name	Description	1	URB_WRITE_OWORD [Default]	URB Oword Write message	
Value	Name	Description						
1	URB_WRITE_OWORD [Default]	URB Oword Write message						



URB Oword Dual Block Read MSD

MSD_UR_OWDB - URB Oword Dual Block Read MSD								
Source:		Read-Only DataPort						
Length Bias:		1						
DWord	Bit	Description						
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header.						
	18	Reserved Format: MBZ Ignored						
	17	Per Slot Offset Format: MHC_PSOP Specifies if per-slot offsets are present and will be added to the Global Offset .						
	16	Reserved Format: MBZ Ignored						
	15	Swizzle Control Format: Opcode <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>URB_INTERLEAVED [Default]</td> <td>Use two URB entries (URB Handle 0 and URB Handle 1).</td> </tr> </tbody> </table>	Value	Name	Description	1	URB_INTERLEAVED [Default]	Use two URB entries (URB Handle 0 and URB Handle 1).
	Value	Name	Description					
	1	URB_INTERLEAVED [Default]	Use two URB entries (URB Handle 0 and URB Handle 1).					
	14:4	Global Offset Format: U11 Specifies the offset, in units of Oword elements, from the start of the URB handle for the access. If Per Slot Offset is set, the global offset is added to those offsets to form the overall offset. Range [0,2047]						
3:0	URB Opcode Format: Opcode <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>URB_READ_OWORD [Default]</td> <td>URB Oword Read message</td> </tr> </tbody> </table>	Value	Name	Description	3	URB_READ_OWORD [Default]	URB Oword Read message	
Value	Name	Description						
3	URB_READ_OWORD [Default]	URB Oword Read message						



URB Oword Dual Block Write MSD

MSD_UW_OWDB - URB Oword Dual Block Write MSD								
Source:		Read-Only DataPort						
Length Bias:		1						
DWord	Bit	Description						
0	19	Header Present Format: MDC_MHR Indicates that the message requires a header.						
	18	Reserved Format: MBZ Ignored						
	17	Per Slot Offset Format: MHC_PSOP Specifies if per-slot offsets are present and will be added to the Global Offset .						
	16	Reserved Format: MBZ Ignored						
	15	Swizzle Control Format: Opcode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>URB_INTERLEAVED [Default]</td> <td>Use two URB entries (URB Handle 0 and URB Handle 1).</td> </tr> </tbody> </table>	Value	Name	Description	1	URB_INTERLEAVED [Default]	Use two URB entries (URB Handle 0 and URB Handle 1).
	Value	Name	Description					
	1	URB_INTERLEAVED [Default]	Use two URB entries (URB Handle 0 and URB Handle 1).					
	14:4	Global Offset Format: U11 Specifies the offset, in units of Oword elements, from the start of the URB handle for the access. If Per Slot Offset is set, the global offset is added to those offsets to form the overall offset. Range [0,2047]						
3:0	URB Opcode Format: Opcode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>URB_WRITE_OWORD [Default]</td> <td>URB Oword Write message</td> </tr> </tbody> </table>	Value	Name	Description	1	URB_WRITE_OWORD [Default]	URB Oword Write message	
Value	Name	Description						
1	URB_WRITE_OWORD [Default]	URB Oword Write message						



VD_PIPELINE_FLUSH

VD_PIPELINE_FLUSH		
Source:		VideoCS
Length Bias:		2
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
		Format: OpCode
	28:27	Pipeline
		Default Value: 2h Media
		Format: OpCode
	26:23	Media Command Opcode
		Default Value: Fh Extended command
		Format: OpCode
	22:21	SubOpcodeA
Default Value: 0h		
Format: OpCode		
20:16	SubOpcodeB	
	Default Value: 0h	
	Format: OpCode	
15:12	Reserved	
	Format: MBZ	
11:0	DWORD_COUNT_n	
	Default Value: 0h Excludes DWord (0)	
	Format: =n	
	Total Length - 2	
1	31:20	Reserved
		Format: MBZ
	19	MFX pipeline command flush
	Format: U1	
18	Reserved	
17	VD-ENC pipeline command flush	
Format: U1		



VD_PIPELINE_FLUSH		
	16	HEVC pipeline command flush Format: U1
	15:5	Reserved Format: MBZ
	4	VD command/message parser Done Format: U1
	3	MFX pipeline Done Format: U1
	2	Reserved
	1	VD-ENC pipeline Done Format: U1
	0	HEVC pipeline Done Format: U1



VEB_DI_IECP

VEB_DI_IECP			
Source:	VideoEnhancementCS		
Length Bias:	2		
<p>The VEB_DI_IECP command causes the VEBOX to start processing the frames specified by VEB_SURFACE_STATE using the parameters specified by VEB_DI_STATE and VEB_IECP_STATE. The processing can start and end at any 64 pixel column in the frame. If Starting X and Ending X are used to split the frame into sections, it should not be split into more than 4 sections.</p> <p>Each VEB_DI_IECP command should be preceded by a VEB_STATE command and the input/output VEB_SURFACE_STATE commands.</p>			
Programming Notes			
When DI is enabled, only the Current Frame skin scores are outputted to the Skin Score Output surface.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:24	Opcode	
		Default Value:	4h VEBOX
		Format:	OpCode
	23:21	SubOpA	
		Default Value:	0h VEB_DI_IECP
		Format:	OpCode
	20:16	SubOpB	
		Default Value:	3h VEB_DI_IECP
Format:		OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Format:	=n Total Length - 2	
	Excludes DWords 0, 1		
	Value	Name	
	14h	[Default]	



VEB_DI_IECP												
1	31:30	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	29:16	Starting X Offset from the beginning of the frame to start processing. Must be a multiple of 64 to guarantee that it starts on a column boundary.										
	15:14	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ											
13:0	Ending X Offset from the beginning of the frame to stop processing. Must be a multiple of 64 or equal to the Surface Width to guarantee that it ends on a column boundary. <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Restriction:Ending_X - Starting_X must be > = 64.</td> </tr> </table>	Programming Notes		Restriction: Ending_X - Starting_X must be > = 64.								
Programming Notes												
Restriction: Ending_X - Starting_X must be > = 64.												
2	31:12	Current Frame Input Address <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> Specifies bits 31:12 of the 4K byte aligned frame buffer address for reading current frame as input to either the DN/DI or IECP stage.	Format:	GraphicsAddress[31:12]								
	Format:	GraphicsAddress[31:12]										
	11	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ											
10:0	Current Frame Surface Control Bits <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS</td> </tr> </table>	Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS									
Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS											
3	31:30	Current Frame Input Surface Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
	Value	Name										
	0	Highest priority										
	1	Second highest priority										
	2	Third highest priority										
3	Lowest priority											
29:16	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
15:0	Current Frame Input Address High <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> Specifies bits 47:32 of the address	Format:	GraphicsAddress[47:32]									
Format:	GraphicsAddress[47:32]											



VEB_DI_IECP												
4	31:12	<p>Previous Frame Input Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies bits 31:12 of the 4K byte aligned frame buffer address for reading the denoised previous frame as input to the DN/DI stage. This field is ignored if DN Enable, DI Enable, and Hot Pixel Filtering Enable are set to 0 (disable).</p>	Format:	GraphicsAddress[31:12]								
	Format:	GraphicsAddress[31:12]										
	11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ											
10:0	<p>Previous Frame Surface Control Bits</p> <table border="1"> <tr> <td>Format:</td> <td>VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS</td> </tr> </table>	Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS									
Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS											
5	31:30	<p>Previous Frame Input Surface Arbitration Priority Control</p> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Highest priority</td> </tr> <tr> <td>1</td> <td>Second highest priority</td> </tr> <tr> <td>2</td> <td>Third highest priority</td> </tr> <tr> <td>3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
	Value	Name										
	0	Highest priority										
	1	Second highest priority										
2	Third highest priority											
3	Lowest priority											
29:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
15:0	<p>Previous Frame Input Address High</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>Specifies bits 47:32 of the address</p>	Format:	GraphicsAddress[47:32]									
Format:	GraphicsAddress[47:32]											
6	31:12	<p>STMM Input Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies bits 31:12 of the 4K byte aligned frame buffer address for reading the STMM / denoise history. This field is ignored if DN Enable, DI Enable, and Hot Pixel Filtering Enable are set to 0 (disable).</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">This field is ignored if Disable Temporal Denoise Filter is set to 1.</td> </tr> </table>	Format:	GraphicsAddress[31:12]	Programming Notes		This field is ignored if Disable Temporal Denoise Filter is set to 1.					
	Format:	GraphicsAddress[31:12]										
	Programming Notes											
	This field is ignored if Disable Temporal Denoise Filter is set to 1.											
11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
10:0	<p>STMM Input Surface Control Bits</p> <table border="1"> <tr> <td>Format:</td> <td>VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS</td> </tr> </table>	Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS									
Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS											



VEB_DI_IECP												
7	31:30	<p>STMM Input Surface Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
		Value	Name									
		0	Highest priority									
		1	Second highest priority									
2	Third highest priority											
3	Lowest priority											
<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ											
<p>STMM Input Address High</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>Specifies bits 47:32 of the address</p>	Format:	GraphicsAddress[47:32]										
Format:	GraphicsAddress[47:32]											
8	31:12	<p>STMM Output Address</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies bits 31:12 of the 4K byte aligned frame buffer address for writing the STMM / denoise history. This field is ignored if DN Enable, DI Enable, and Hot Pixel Filtering Enable are set to 0 (disable).</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>This field is ignored if Disable Temporal Denoise Filter is set to 1.</td> </tr> </tbody> </table>	Format:	GraphicsAddress[31:12]	Programming Notes	This field is ignored if Disable Temporal Denoise Filter is set to 1.						
		Format:	GraphicsAddress[31:12]									
	Programming Notes											
	This field is ignored if Disable Temporal Denoise Filter is set to 1.											
<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ											
11												
10:0	<p>STMM Output Surface Control Bits</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Format:</td> <td>VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS</td> </tr> </table>	Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS									
Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS											
9	31:30	<p>STMM Output Surface Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
		Value	Name									
		0	Highest priority									
		1	Second highest priority									
		2	Third highest priority									
3	Lowest priority											
<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ											
<p>STMM Output Address High</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>Specifies bits 47:32 of the address</p>	Format:	GraphicsAddress[47:32]										
Format:	GraphicsAddress[47:32]											



VEB_DI_IECP												
10	31:12	<p>Denoisied Current Frame Output Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies bits 31:12 of the 4K byte aligned frame buffer address for writing the current frame after the DN stage. This field is ignored if both DN Enable and Hot Pixel Filtering Enable are set to 0 (disable).</p>	Format:	GraphicsAddress[31:12]								
	Format:	GraphicsAddress[31:12]										
	11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ											
10:0	<p>Denoisied Current Output Surface Control Bits</p> <table border="1"> <tr> <td>Format:</td> <td>VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS</td> </tr> </table>	Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS									
Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS											
11	31:30	<p>Denoisied Current Output Surface Arbitration Priority Control</p> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Highest priority</td> </tr> <tr> <td>1</td> <td>Second highest priority</td> </tr> <tr> <td>2</td> <td>Third highest priority</td> </tr> <tr> <td>3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
	Value	Name										
	0	Highest priority										
	1	Second highest priority										
2	Third highest priority											
3	Lowest priority											
29:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
15:0	<p>Denoisied Current Frame Output Address High</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>Specifies bits 47:32 of the address</p>	Format:	GraphicsAddress[47:32]									
Format:	GraphicsAddress[47:32]											
12	31:12	<p>Current Frame Output Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies bits 31:12 of the 4K byte aligned frame buffer address for writing the current frame output. The output is from DN/DI if IECP is disabled, or from IECP if enabled.</p>	Format:	GraphicsAddress[31:12]								
	Format:	GraphicsAddress[31:12]										
	11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ											
10:0	<p>Current Frame Output Surface Control Bits</p> <table border="1"> <tr> <td>Format:</td> <td>VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS</td> </tr> </table>	Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS									
Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS											
13	31:30	<p>Current Frame Output Surface Arbitration Priority Control</p> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> </tbody> </table>	Value	Name								
	Value	Name										



VEB_DI_IECP											
	<table border="1"> <tr> <td>0</td> <td>Highest priority</td> </tr> <tr> <td>1</td> <td>Second highest priority</td> </tr> <tr> <td>2</td> <td>Third highest priority</td> </tr> <tr> <td>3</td> <td>Lowest priority</td> </tr> </table>	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority		
0	Highest priority										
1	Second highest priority										
2	Third highest priority										
3	Lowest priority										
	29:16 Reserved Format: <table border="1"><tr><td> </td><td>MBZ</td></tr></table>		MBZ								
	MBZ										
	15:0 Current Frame Output Address High Format: <table border="1"><tr><td> </td><td>GraphicsAddress[47:32]</td></tr></table> Specifies bits 47:32 of the address		GraphicsAddress[47:32]								
	GraphicsAddress[47:32]										
14	31:12 Previous Frame Output Address Format: <table border="1"><tr><td> </td><td>GraphicsAddress[31:12]</td></tr></table> Specifies bits 31:12 of the 4K byte aligned frame buffer address for writing the previous frame output. This field is ignored if DI Enable is set to 0 (disable).		GraphicsAddress[31:12]								
	GraphicsAddress[31:12]										
	11 Reserved Format: <table border="1"><tr><td> </td><td>MBZ</td></tr></table>		MBZ								
	MBZ										
	10:0 Previous Frame Output Surface Control Bits Format: <table border="1"><tr><td> </td><td>VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS</td></tr></table>		VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS								
	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS										
15	31:30 Previous Frame Output Surface Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Highest priority</td> </tr> <tr> <td>1</td> <td>Second highest priority</td> </tr> <tr> <td>2</td> <td>Third highest priority</td> </tr> <tr> <td>3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
Value	Name										
0	Highest priority										
1	Second highest priority										
2	Third highest priority										
3	Lowest priority										
	29:16 Reserved Format: <table border="1"><tr><td> </td><td>MBZ</td></tr></table>		MBZ								
	MBZ										
	15:0 Previous Frame Output Address High Format: <table border="1"><tr><td> </td><td>GraphicsAddress[47:32]</td></tr></table> Specifies bits 47:32 of the address		GraphicsAddress[47:32]								
	GraphicsAddress[47:32]										
16	31:12 Statistics Output Address Format: <table border="1"><tr><td> </td><td>GraphicsAddress[31:12]</td></tr></table> Specifies bits 31:12 of the 4K byte aligned frame buffer address for writing block level FMD and DN statistics as well as the frame level ACE histogram and FMD frame level statistics.		GraphicsAddress[31:12]								
	GraphicsAddress[31:12]										



VEB_DI_IECP												
	11	Reserved Format: _____ MBZ										
	10:0	Statistics Output Surface Control Bits Format: VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS										
17	31:30	Statistics Output Surface Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
		Value	Name									
		0	Highest priority									
		1	Second highest priority									
		2	Third highest priority									
3	Lowest priority											
29:16	Reserved Format: _____ MBZ											
15:0	Statistics Output Address High Format: _____ GraphicsAddress[47:32] Specifies bits 47:32 of the address											
18	31:12	Alpha/Vignette Correction Address Format: _____ GraphicsAddress[31:12] Specifies bits 31:12 of the 4K byte aligned frame buffer address for reading the Alpha surface or vignette correction.										
	11	Reserved Format: _____ MBZ										
	10:0	Alpha/Vignette Control Bits Format: VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS										
19	31:30	Alpha/Vignette Correction Surface Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
		Value	Name									
		0	Highest priority									
		1	Second highest priority									
		2	Third highest priority									
3	Lowest priority											
29:16	Reserved Format: _____ MBZ											
15:0	Alpha/Vignette Correction Address High Format: _____ GraphicsAddress[47:32] Specifies bits 47:32 of the address											



VEB_DI_IECP														
20	31:12	<p>LACE/ACE/RGB Histogram Output Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies bits 31:12 of the 4K byte aligned address for writing the LACE histograms, or ACE histogram, and/or RGB histogram.</p>	Format:	GraphicsAddress[31:12]										
	Format:	GraphicsAddress[31:12]												
	11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ													
10:0	<p>LACE/ACE/RGB Histogram Control Bits</p> <table border="1"> <tr> <td>Format:</td> <td>VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS</td> </tr> </table> <p>Please refer to the Surface Control Bits Table below.</p>	Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS											
Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS													
21	31:30	<p>LACE/ACE/RGB Histogram Surface Arbitration Priority Control</p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Format:	U2	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
	Format:	U2												
	Value	Name												
	0	Highest priority												
1	Second highest priority													
2	Third highest priority													
3	Lowest priority													
29:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ											
Format:	MBZ													
15:0	<p>LACE/ACE/RGB Histogram Output Address High</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>Bits 47:32 of address.</p>	Format:	GraphicsAddress[47:32]											
Format:	GraphicsAddress[47:32]													
22	31:12	<p>Skin Score Output Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> <p>Specifies bits 31:12 of the 4K byte aligned address for writing the Skin score output in case enabled.</p>	Format:	GraphicsAddress[31:12]										
	Format:	GraphicsAddress[31:12]												
	11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ													
10:0	<p>Skin Score Output Control Bits</p> <table border="1"> <tr> <td>Format:</td> <td>VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS</td> </tr> </table> <p>Please refer to the Surface Control Bits Table below.</p>	Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS											
Format:	VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS													



VEB_DI_IECP												
23	31:30	<p>Skin Score Output Surface Arbitration Priority Control This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Lowest priority</td> </tr> </tbody> </table>	Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
		Value	Name									
		0	Highest priority									
		1	Second highest priority									
		2	Third highest priority									
3	Lowest priority											
29:16	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
15:0	<p>Skin Score Output Address High</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>Bits 47:32 of address</p>	Format:	GraphicsAddress[47:32]									
Format:	GraphicsAddress[47:32]											



VEBOX_STATE

VEBOX_STATE		
Source:	VideoEnhancementCS	
Length Bias:	2	
Description		
<p>This command controls the internal functions of the VEBOX. This command has a set of indirect state buffers:</p> <ul style="list-style-type: none"> • DN/DI state • IECP general state • IECP Gamut Expansion/Compression state • IECP Gamut Vertex Table state • Capture Pipe state 		
: Adds the LACE LUT Table as an indirect state buffer.		
DWord	Bit	Description
0	31:29	Command Type
		Default Value: 3h PARALLEL_VIDEO_PIPE
	Format: OpCode	
	28:27	Pipeline
		Default Value: 2h Media
	Format: OpCode	
	26:24	Command OpCode
		Default Value: 4h VEBOX
	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	



VEBOX_STATE																				
	11:0	DWord Length																		
		Format: =n Total Length - 2																		
		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Eh</td> <td></td> <td>(Excludes DWords 0, 1)</td> </tr> </tbody> </table>	Value	Name	Description	Eh		(Excludes DWords 0, 1)												
Value	Name	Description																		
Eh		(Excludes DWords 0, 1)																		
1	31:25	State Surface Control Bits All Indirect state buffers use state surface control bits, only exception being 3D LUT state buffer for which the state surface control bits are tied to 0. See definition under "VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS " Bits[6:0] is only used.																		
	24:23	Reserved Format: MBZ																		
	22	Reserved Format: MBZ																		
	21	Forward Gamma Correction Enable Format: Enable <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">Single Pipe IECP Enable must also be set if this is enabled.</td> </tr> </tbody> </table>	Programming Notes		Single Pipe IECP Enable must also be set if this is enabled.															
	Programming Notes																			
	Single Pipe IECP Enable must also be set if this is enabled.																			
	20	Reserved Format: MBZ																		
	19	Single Pipe Enable <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">Indicates that the Capture Pipe features that only exist in a single pipe can be enabled.</td> </tr> <tr> <td colspan="2">This bit must be set if any of the following features are enabled: Demosaic Denoise with one of the RGBA input formats IECP only mode with Forward Gamma Correction enabled with RGB input formats (All other modes are not supported in single pipe)</td> </tr> <tr> <td colspan="2">This bit must be set if any of the following features are enabled: IECP only mode with either Color Correction Matrix, Forward Gamma Correction, Front-End CSC enabled.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>Enable</td> </tr> <tr> <td style="text-align: center;">0</td> <td>Default [Default]</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">Note that the pixel throughput is 1/2 when this mode is selected. The Global IECP Enable must also be set.</td> </tr> </tbody> </table>	Description		Indicates that the Capture Pipe features that only exist in a single pipe can be enabled.		This bit must be set if any of the following features are enabled: Demosaic Denoise with one of the RGBA input formats IECP only mode with Forward Gamma Correction enabled with RGB input formats (All other modes are not supported in single pipe)		This bit must be set if any of the following features are enabled: IECP only mode with either Color Correction Matrix, Forward Gamma Correction, Front-End CSC enabled.		Value	Name	1	Enable	0	Default [Default]	Programming Notes		Note that the pixel throughput is 1/2 when this mode is selected. The Global IECP Enable must also be set.	
	Description																			
	Indicates that the Capture Pipe features that only exist in a single pipe can be enabled.																			
This bit must be set if any of the following features are enabled: Demosaic Denoise with one of the RGBA input formats IECP only mode with Forward Gamma Correction enabled with RGB input formats (All other modes are not supported in single pipe)																				
This bit must be set if any of the following features are enabled: IECP only mode with either Color Correction Matrix, Forward Gamma Correction, Front-End CSC enabled.																				
Value	Name																			
1	Enable																			
0	Default [Default]																			
Programming Notes																				
Note that the pixel throughput is 1/2 when this mode is selected. The Global IECP Enable must also be set.																				



VEBOX_STATE															
18	<p>Disable Temporal Denoise Filter If set this bit will force the denoise filter to only use the spatial filter. This will eliminate the read of the previous denoise surface and STMM/Denoise History surface and the write of the current denoised surface and STMM/Denoise History surface.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> <p>The Global IECP Enable or Demosaic Enable must be set along with this bit. This bit must be set if the input to Denoise is RGB. This bit must not be set if the Deinterlacer is enabled. This bit must be clear if both DN Enable=0 and Hot Pixel Filtering Enable=0 This bit must be set if Hot Pixel Filtering Enable=1 and both DN and DI are disabled</p> </td> </tr> </tbody> </table>	Programming Notes		<p>The Global IECP Enable or Demosaic Enable must be set along with this bit. This bit must be set if the input to Denoise is RGB. This bit must not be set if the Deinterlacer is enabled. This bit must be clear if both DN Enable=0 and Hot Pixel Filtering Enable=0 This bit must be set if Hot Pixel Filtering Enable=1 and both DN and DI are disabled</p>											
Programming Notes															
<p>The Global IECP Enable or Demosaic Enable must be set along with this bit. This bit must be set if the input to Denoise is RGB. This bit must not be set if the Deinterlacer is enabled. This bit must be clear if both DN Enable=0 and Hot Pixel Filtering Enable=0 This bit must be set if Hot Pixel Filtering Enable=1 and both DN and DI are disabled</p>															
17	<p>Disable Encoder Statistics If set this bit will disable writing the per block Encoder statistics. The memory format is not changed, so the area set aside for these statistics will still be there.</p>														
16	<p>LACE Correction Enable This bit enables the correction of the image according to the local ACE LUT tables. This is independent from the enable for the collection of LACE histograms.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> <p>LACE correction is only enabled if both this bit and the Global IECP Enable are set. The ACE Enable bit should also be set if this bit is set, since ACE correction can be used for part of the luma range instead of LACE.</p> </td> </tr> </tbody> </table>	Programming Notes		<p>LACE correction is only enabled if both this bit and the Global IECP Enable are set. The ACE Enable bit should also be set if this bit is set, since ACE correction can be used for part of the luma range instead of LACE.</p>											
Programming Notes															
<p>LACE correction is only enabled if both this bit and the Global IECP Enable are set. The ACE Enable bit should also be set if this bit is set, since ACE correction can be used for part of the luma range instead of LACE.</p>															
15:14	<p>Single Slice VEBOX Enable For products that have 2 entire VEBOXes that automatically split the frame, this enable emulates a 1 VEBOX product, running at 1/2 speed and only outputting a single set of per command statistics.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; background-color: #e6f2ff;">Value</th> <th style="text-align: center; background-color: #e6f2ff;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Both Slices Enabled</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Slice 0 Enabled</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Slice 1 Enables</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Reserved</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> <p>This field must be set if 00b for products that don't have 2 slices.</p> </td> </tr> </tbody> </table>	Value	Name	00b	Both Slices Enabled	01b	Slice 0 Enabled	10b	Slice 1 Enables	11b	Reserved	Programming Notes		<p>This field must be set if 00b for products that don't have 2 slices.</p>	
Value	Name														
00b	Both Slices Enabled														
01b	Slice 0 Enabled														
10b	Slice 1 Enables														
11b	Reserved														
Programming Notes															
<p>This field must be set if 00b for products that don't have 2 slices.</p>															
13	<p>Hot Pixel Filtering Enable Enables hot pixel detection/filtering.</p>														
12	<p>Alpha Plane Enable Enables the reading of an independent Alpha plane. Mutually exclusive with Vignette Enable. If Alpha from State Select is set it overrides this bit.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> <p>IECP must also be enabled and output format must have alpha if this bit is enabled. Should be 0 if Alpha from State Select is 1.</p> </td> </tr> </tbody> </table>	Programming Notes		<p>IECP must also be enabled and output format must have alpha if this bit is enabled. Should be 0 if Alpha from State Select is 1.</p>											
Programming Notes															
<p>IECP must also be enabled and output format must have alpha if this bit is enabled. Should be 0 if Alpha from State Select is 1.</p>															



VEBOX_STATE																									
11	<p>Vignette Enable Enables Vignette Correction surface read and correction in IECP. Mutually exclusive with Alpha Plane Enable.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">Demosaic must also be enabled if this bit is enabled.</td> </tr> </table>	Programming Notes		Demosaic must also be enabled if this bit is enabled.																					
Programming Notes																									
Demosaic must also be enabled if this bit is enabled.																									
10	<p>Demosaic Enable The Demosaic will be used, and White balance statistics will be gathered. The Capture Pipe State Table will be read. This bit is mutually exclusive with DI Enable.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">IECP must also be enabled if this bit is enabled.</td> </tr> </table>	Programming Notes		IECP must also be enabled if this bit is enabled.																					
Programming Notes																									
IECP must also be enabled if this bit is enabled.																									
9:8	<p>DI Output Frames Indicates which frames to output in DI mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>Output Both Frames</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Output Previous Frame Only</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Output Current Frame Only</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">Field is ignored if DI Enable = 0. If Previous Frame Only or Current Frame Only are selected, then the LACE Single Histogram Set must not try to collect a histogram from the disabled frame. Field must be programmed to 10 (Output Current Frame Only) for DI First Frame.</td> </tr> </table>	Value	Name	00b	Output Both Frames	01b	Output Previous Frame Only	10b	Output Current Frame Only	Programming Notes		Field is ignored if DI Enable = 0. If Previous Frame Only or Current Frame Only are selected, then the LACE Single Histogram Set must not try to collect a histogram from the disabled frame. Field must be programmed to 10 (Output Current Frame Only) for DI First Frame.													
Value	Name																								
00b	Output Both Frames																								
01b	Output Previous Frame Only																								
10b	Output Current Frame Only																								
Programming Notes																									
Field is ignored if DI Enable = 0. If Previous Frame Only or Current Frame Only are selected, then the LACE Single Histogram Set must not try to collect a histogram from the disabled frame. Field must be programmed to 10 (Output Current Frame Only) for DI First Frame.																									
7	<p>444 -> 422 Downsample Method</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>Average horizontally aligned chromas</td> </tr> <tr> <td style="text-align: center;">0</td> <td>Drop right chroma of the pair [Default]</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr> <th colspan="3" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td style="text-align: center;">444->422</td> <td style="text-align: center;">422->420</td> <td style="text-align: center;">Description</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>No averaging, only down sampling</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Not Supported</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Only Horizontal averaging</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Horizontal and Vertical averaging</td> </tr> </table>	Value	Name	1	Average horizontally aligned chromas	0	Drop right chroma of the pair [Default]	Programming Notes			444->422	422->420	Description	0	0	No averaging, only down sampling	0	1	Not Supported	1	0	Only Horizontal averaging	1	1	Horizontal and Vertical averaging
Value	Name																								
1	Average horizontally aligned chromas																								
0	Drop right chroma of the pair [Default]																								
Programming Notes																									
444->422	422->420	Description																							
0	0	No averaging, only down sampling																							
0	1	Not Supported																							
1	0	Only Horizontal averaging																							
1	1	Horizontal and Vertical averaging																							
6	<p>422 -> 420 Downsample Method</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>Average vertically aligned chromas</td> </tr> <tr> <td style="text-align: center;">0</td> <td>Drop lower chroma of the pair [Default]</td> </tr> </tbody> </table>	Value	Name	1	Average vertically aligned chromas	0	Drop lower chroma of the pair [Default]																		
Value	Name																								
1	Average vertically aligned chromas																								
0	Drop lower chroma of the pair [Default]																								



VEBOX_STATE

		Programming Notes
		To enable averaging in case of 420 (NV12/P016) output formats, 444->422 and 422->420 should be set.
5	DN/DI First Frame	
	Format:	Enable
	Indicates that this is the first frame of the stream, so previous clean is not available.	
	Value	Name
	0	Not first field; previous clean surface state is valid
	1	First field; previous clean surface state is invalid
	Programming Notes	
	If both DN and DI are disabled, this bit must be 0.	
4	DI Enable	
	Format:	Enable
	Deinterlacer is bypassed if this is disabled: the output is the same as the input (same as a 2:2 cadence).	
	FMD and STMM are not calculated and the values in the response message are 0.	
	Value	Name
	0	Do not calculate DI
	1	Calculate DI
3	DN Enable	
	Format:	Enable
	Denoise is bypassed if this is low - BNE is still calculated and output, but the denoised fields are not. VDI does not read in the denoised previous frame but uses the pointer for the original previous frame.	
	Value	Name
	0	Do not denoise frame
	1	Denoise frame
	Programming Notes	
	If DN and/or Hotpixel are the only functions enabled then the only output is the Denoised Output which is the same surface format as the input.	
	To get a format conversion with DN only, enable the Global IECP bit, but disable all the individual functions. The IECP output uses the output surface format.	
	If DN is used with RGB then the Global IECP Enable must also be	
2	Global IECP Enable	
	Indicates if any of the IECP features is enabled.	
	If this is disabled then no state will be read from any of the state pointers. If set then the IECP state will be read.	



VEBOX_STATE				
	1	<p>Color Gamut Compression Enable Indicates if the Gamut Compression feature is enabled. If set then the Gamut State will be read. VEB_VERTEXTABLE_STATE is only needed if this bit is set.</p>		
	0	<p>Color Gamut Expansion Enable Indicates if the Gamut Expansion feature is enabled. If set then the Gamut State will be read.</p>		
2	31:12	<p>DN/DI State Pointer Low</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicAddress[31:12]</td> </tr> </table> <p>Bits 31:12 of the starting address of the DN/DI State buffer. This points to a buffer containing the 10 Dwords of the DN/DI state.</p>	Format:	GraphicAddress[31:12]
	Format:	GraphicAddress[31:12]		
11:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
3	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
15:0	<p>DN/DI State Pointer High</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicAddress[47:32]</td> </tr> </table> <p>Bits 47:32 of the starting address of the DN/DI State Buffer.</p>	Format:	GraphicAddress[47:32]	
Format:	GraphicAddress[47:32]			
4	31:12	<p>IECP State Pointer Low</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicAddress[31:12]</td> </tr> </table> <p>Bits 31:12 of the starting address of the IECP State buffer. This points to a buffer containing the 64 Dwords of IECP state.</p>	Format:	GraphicAddress[31:12]
	Format:	GraphicAddress[31:12]		
11:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
5	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
15:0	<p>IECP State Pointer High</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicAddress[47:32]</td> </tr> </table> <p>Bits 47:32 of the starting address of the IECP State Buffer Table.</p>	Format:	GraphicAddress[47:32]	
Format:	GraphicAddress[47:32]			
6	31:12	<p>Gamut State Pointer Low</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicAddress[31:12]</td> </tr> </table> <p>Bits 31:12 of the starting address of the Gamut State buffer. This points to a buffer containing the 42 Dwords of Gamut Compression / Gamut Expansion state.</p>	Format:	GraphicAddress[31:12]
	Format:	GraphicAddress[31:12]		
11:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			



VEBOX_STATE		
7	31:16	Reserved Format: _____ MBZ
	15:0	Gamut State Pointer High Format: _____ GraphicAddress[47:32] Bits 47:32 of the starting address of the Gamut State Buffer.
8	31:12	Vertex Table State Pointer Low Format: _____ GraphicAddress[31:12] Bits 31:12 of the starting address of the Vertex Table. This points to a buffer containing the 512 Dwords of the Gamut Compression Vertex Table.
	11:0	Reserved Format: _____ MBZ
9	31:16	Reserved Format: _____ MBZ
	15:0	Vertex Table State Pointer High Format: _____ GraphicAddress[47:32] Bits 47:32 of the starting address of the Vertex State Buffer.
10	31:12	Capture Pipe State Pointer Low Format: _____ GraphicAddress[31:12] Bits 31:12 of the starting address of the Capture Pipe State Table. This points to a buffer containing the X Dwords of the Capture Pipe State.
	11:0	Reserved Format: _____ MBZ
11	31:16	Reserved Format: _____ MBZ
	15:0	Capture Pipe State Pointer High Format: _____ GraphicAddress[47:32] Bits 47:32 of the starting address of the Capture Pipe State Table.
12	31:12	LACE LUT Table State Pointer Low Format: _____ GraphicAddress[31:12] Bits [31:12] of the starting address of the LACE Look-up Tables.
	11:0	Reserved Format: _____ MBZ



VEBOX_STATE												
13	31:30	Arbitration Priority Control - For LACE LUT Format: U2 This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.										
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Highest priority</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Second highest priority</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Third highest priority</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Lowest priority</td> </tr> </tbody> </table>		Value	Name	0	Highest priority	1	Second highest priority	2	Third highest priority	3	Lowest priority
	Value	Name										
	0	Highest priority										
	1	Second highest priority										
2	Third highest priority											
3	Lowest priority											
29:16	Reserved Format: MBZ											
15:0	LACE LUT Table State Pointer High Format: GraphicAddress[47:32] Bits [47:32] of the starting address of the LACE Look-up Tables.											
14..15	63:12	Gamma Correction Values Address Format: GraphicsAddress63-12 Specifies the 4K byte aligned address reading the Gamma Correction Values in case enabled.										
	11:0	Reserved Format: MBZ										



VEBOX_SURFACE_STATE

VEBOX_SURFACE_STATE	
Source:	VideoEnhancementCS
Length Bias:	2
Description	
The input and output data containers accessed are called "surfaces". Surface state is sent to VEBOX via an inline state command rather than using binding tables. SURFACE_STATE contains the parameters defining each surface to be accessed, including its size, format, and offsets to its subsurfaces. The surface's base address is in the execution command. Despite having multiple input and output surfaces, we limit the number of surface states to one for input surfaces and one for output surfaces. The other surfaces are derived from the input/output surface states.	
The Current Frame Input surface uses the Input SURFACE_STATE	
The Previous Denoised Input surface uses the Input SURFACE_STATE. (For 16-bit Bayer pattern inputs this will be 16-bit.)	
The Current Denoised Output surface uses the Input SURFACE_STATE. (For 16-bit Bayer pattern inputs this will be 16-bit.)	
The STMM/Noise History Input surface uses the Input SURFACE_STATE with Tile-Y and Width/Height a multiple of 4.	
The STMM/Noise History Output surface uses the Input SURFACE_STATE with Tile-Y and Width/Height a multiple of 4.	
The Current Deinterlaced/IECP Frame Output surface uses the Output SURFACE_STATE.	
The Previous Deinterlaced/IECP Frame Output surface uses the Output SURFACE_STATE.	
The FMD per block output / per Frame Output surface uses the Linear SURFACE_STATE (see note below).	
The Alpha surface uses the Linear A8 SURFACE_STATE with Width/Height equal to Input Surface. Pitch is width rounded to next 64.	
The Skin Score surface uses the Output SURFACE_STATE.	
The STMM height is the same as the Input Surface height except when the input Surface Format is Bayer Pattern and the Bayer Pattern Offset is 10 or 11, in which case the height is the input height + 4. For Bayer pattern inputs when the Bayer Pattern Offset is 10 or 11, the Current Denoised Output/Previous Denoised Input will also have a height which is the input height + 4. For Bayer pattern inputs only the Current Denoised Output/Previous Denoised Input are in Tile-Y.	
The linear surface for FMD statistics is linear (not tiled). The height of the per block statistics is (Input Height + 3)/4 - the Input Surface height in pixels is rounded up to the next even 4 and divided by 4. The width of the per block section in bytes is equal to the width of the Input Surface in pixels rounded up to the next 16 bytes. The pitch of the per block section in bytes is equal to the width of the Input Surface in pixels rounded up to the next 64 bytes.	
The STMM surfaces must be identical to the Input surface except for the tiling mode must be Tile-Y and the pitch is specified in DW7. The pitch for the Current Denoised Output/Previous Denoised Input is specified in DW7. The width and height must be a multiple of 4 rounded up from the input height.	



VEBOX_SURFACE_STATE

The Vignette Correction surface uses the Linear 16-bit SURFACE_STATE with :

$$\text{Width} = (\text{Ceil}(\text{Image Width} / 4) + 1) * 4$$

$$\text{Height} = \text{Ceil}(\text{Image Height} / 4) + 1$$

Pitch in bytes is (vignette width *2) rounded to the next 64

Programming Notes

VEBOX may write to memory between the surface width and the surface pitch for output surfaces.

VEBOX can support a frame level X/Y offset which allows processing of 2 side-by-side frames for certain 3D video formats.

The X/Y Offset for Frame state applies only to the Current Frame Input and the Current Deinterlaced/IECP Frame Output and Previous Deinterlaced/IECP Frame Output. The statistics surfaces, the denoise feedback surfaces and the alpha/vignette surfaces have no X/Y offsets.

For 8bit Alpha input, when converted to 16bit output, the 8 bit alpha value is replicated to both the upper and lower 8 bits to form the 16 bit alpha value.

Skin Score Output Surface uses the same tiling format as the Output surface.

DWord	Bit	Description					
0	31:29	Command Type					
		Default Value: 3h PARALLEL_VIDEO_PIPE					
		Format: OpCode					
	28:27	Media Command Pipeline					
		Default Value: 2h Media					
		Format: OpCode					
	26:24	Media Command OpCode					
		Default Value: 4h VEBOX					
		Format: OpCode					
	23:21	SubOpcode A					
		Default Value: 0h VEBOX					
		Format: OpCode					
	20:16	SubOpcode B					
		Default Value: 0h VEBOX					
Format: OpCode							
15:12	Reserved						
	Format: MBZ						
11:0	DWord Length	Format: =n Total Length - 2					
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">7h</td> <td>DWORD_COUNT_n [Default]</td> <td>(Excludes DWords 0, 1)</td> </tr> </tbody> </table>	Value	Name	Description	7h	DWORD_COUNT_n [Default]
	Value	Name	Description				
7h	DWORD_COUNT_n [Default]	(Excludes DWords 0, 1)					



VEBOX_SURFACE_STATE				
1	31:1	Reserved		
	Format:		MBZ	
0	Surface Identification			
	Specifies which set of surfaces this command refers to:			
	Value	Name		
	1	Output surface (all except the Denoised Current output surface)		
0	Input surface and Denoised Current Output Surface			
2	31:18	Height		
	Format:		U14	
	This field specifies the height of the surface in units of pixels. For PLANAR surface formats, this field indicates the height of the Y (luma) plane.			
	Value	Name	Description	Exists If
	[15, 16383]		representing heights [16,16384]	
	[15, 8191]			//Scalar Enabled - For Input surface only
	[63, 2047]			//Scalar + SFC Enabled - For Input surface only
	Programming Notes			
	<p>Height (field value + 1) must be a multiple of 2 for PLANAR_420 surfaces.</p> <p>Height (field value + 1) must be a multiple of 2 when the deinterlace function is enabled (field mode) or when the denoise function is enabled with Progressive DN = 0. It must be a multiple of 4 when interleaved deinterlace/denoise and PLANAR_420 are both being used.</p> <p>VEBOX supports a minimum height of 16.</p> <p>Height (field value + 1) must be a multiple of 2 for Bayer surfaces.</p>			
	17:4	Width		
Format:		U14		
This field specifies the width of the surface in units of pixels. For PLANAR surface formats, this field indicates the width of the Y (luma) plane.				
Value	Name	Description	Exists If	
[63,16383]		representing widths [64,16384]		
[63,8191]			//Scalar Enabled - For Input surface only	
[63,2047]			//Scalar and SFC Enabled - For Input Surface only	
Programming Notes				
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).				



VEBOX_SURFACE_STATE																																																							
		<p>Width (field value + 1) must be a multiple of 2 for PLANAR_420, PLANAR_422, and all YCRCB_* surfaces, and must be a multiple of 4 for PLANAR_411 surfaces. VEBOX supports a minimum width of 64</p>																																																					
	3:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																																																			
Format:	MBZ																																																						
3	31:28	<p>Surface Format</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>YCRCB_NORMAL</td><td></td></tr> <tr><td>1</td><td>YCRCB_SWAPUVY</td><td></td></tr> <tr><td>2</td><td>YCRCB_SWAPUV</td><td></td></tr> <tr><td>3</td><td>YCRCB_SWAPY</td><td></td></tr> <tr><td>4</td><td>PLANAR_420_8</td><td>NV12 with Interleave Chroma set</td></tr> <tr><td>5</td><td>PACKED_444A_8</td><td>IECP input/output only</td></tr> <tr><td>6</td><td>PACKED_422_16</td><td>IECP input/output only</td></tr> <tr><td>7</td><td>R10G10B10A2_UNORM / R10G10B10A2_UNORM_SRGB</td><td>IECP output only</td></tr> <tr><td>8</td><td>R8G8B8A8_UNORM / R8G8B8A8_UNORM_SRGB</td><td>Hot Pixel/Denoise and/or IECP input/output</td></tr> <tr><td>9</td><td>PACKED_444_16</td><td>IECP input/output only</td></tr> <tr><td>10</td><td>PLANAR_422_16</td><td>IECP input/output only</td></tr> <tr><td>11</td><td>Y8_UNORM</td><td></td></tr> <tr><td>12</td><td>PLANAR_420_16</td><td>IECP input/output only</td></tr> <tr><td>13</td><td>R16G16B16A16</td><td>Hot Pixel/Denoise and/or IECP input/output</td></tr> <tr><td>14</td><td>Bayer pattern</td><td>Demosaic input only</td></tr> <tr><td>15</td><td>Y16_UNORM</td><td>Denoise/IECP input/output</td></tr> </tbody> </table>	Format:	U4	Value	Name	Description	0	YCRCB_NORMAL		1	YCRCB_SWAPUVY		2	YCRCB_SWAPUV		3	YCRCB_SWAPY		4	PLANAR_420_8	NV12 with Interleave Chroma set	5	PACKED_444A_8	IECP input/output only	6	PACKED_422_16	IECP input/output only	7	R10G10B10A2_UNORM / R10G10B10A2_UNORM_SRGB	IECP output only	8	R8G8B8A8_UNORM / R8G8B8A8_UNORM_SRGB	Hot Pixel/Denoise and/or IECP input/output	9	PACKED_444_16	IECP input/output only	10	PLANAR_422_16	IECP input/output only	11	Y8_UNORM		12	PLANAR_420_16	IECP input/output only	13	R16G16B16A16	Hot Pixel/Denoise and/or IECP input/output	14	Bayer pattern	Demosaic input only	15	Y16_UNORM	Denoise/IECP input/output
Format:	U4																																																						
Value	Name	Description																																																					
0	YCRCB_NORMAL																																																						
1	YCRCB_SWAPUVY																																																						
2	YCRCB_SWAPUV																																																						
3	YCRCB_SWAPY																																																						
4	PLANAR_420_8	NV12 with Interleave Chroma set																																																					
5	PACKED_444A_8	IECP input/output only																																																					
6	PACKED_422_16	IECP input/output only																																																					
7	R10G10B10A2_UNORM / R10G10B10A2_UNORM_SRGB	IECP output only																																																					
8	R8G8B8A8_UNORM / R8G8B8A8_UNORM_SRGB	Hot Pixel/Denoise and/or IECP input/output																																																					
9	PACKED_444_16	IECP input/output only																																																					
10	PLANAR_422_16	IECP input/output only																																																					
11	Y8_UNORM																																																						
12	PLANAR_420_16	IECP input/output only																																																					
13	R16G16B16A16	Hot Pixel/Denoise and/or IECP input/output																																																					
14	Bayer pattern	Demosaic input only																																																					
15	Y16_UNORM	Denoise/IECP input/output																																																					
	27	<p>Interleave Chroma</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This field indicates that the chroma fields are interleaved in a single plane rather than stored as two separate planes. This field is only used for PLANAR surface formats.</p>	Format:	Enable																																																			
Format:	Enable																																																						
	26:25	<p>Bayer Pattern Offset</p> <p>Specifies the starting pixel offset for the Bayer pattern used for Capture Pipe.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Pixel at X=0, Y=0 is Blue</td> </tr> </tbody> </table>	Value	Name	00b	Pixel at X=0, Y=0 is Blue																																																	
Value	Name																																																						
00b	Pixel at X=0, Y=0 is Blue																																																						



VEBOX_SURFACE_STATE														
	01b	Pixel at X=0, Y=0 is Red												
	10b	Pixel at X=0, Y=0 is Green, Pixel at X=1, Y=0 is Red												
	11b	Pixel at X=0, Y=0 is Green, Pixel at X=1, Y=0 is Blue												
24	Bayer Pattern Format Specifies the format of the Bayer Pattern: <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>8-bit input at a 8-bit stride</td> </tr> <tr> <td>1b</td> <td>16-bit input at a 16-bit stride</td> </tr> </tbody> </table>		Value	Name	0b	8-bit input at a 8-bit stride	1b	16-bit input at a 16-bit stride						
Value	Name													
0b	8-bit input at a 8-bit stride													
1b	16-bit input at a 16-bit stride													
23:21	Reserved Format: _____ MBZ													
20	Reserved Format: _____ MBZ													
19:3	Surface Pitch Format: _____ U17 pitch in (Bytes - 1) This field specifies the surface pitch in (#Bytes - 1): <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td>[63, 131071]</td> <td>For other linear surfaces</td> <td>[64B, 128KB]</td> </tr> <tr> <td>[511, 131071]</td> <td>For X-tiled surface</td> <td>[512B, 128KB] = [1tile, 256 tiles]</td> </tr> <tr> <td>[127, 131071]</td> <td>For Y-tiled surfaces</td> <td>[128B,128KB] = [1 tile, 1024 tiles]</td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p style="text-align: center; color: blue; font-weight: bold;">Programming Notes</p> <p>For tiled surfaces, the pitch must be a multiple of the tile width. For linear surfaces, the pitch must be a multiple of 64. If Half Pitch for Chroma is set, this field must be a multiple of two tile widths for tiled surfaces, or a multiple of 2 bytes for linear surfaces.</p> </div>		Value	Name	Description	[63, 131071]	For other linear surfaces	[64B, 128KB]	[511, 131071]	For X-tiled surface	[512B, 128KB] = [1tile, 256 tiles]	[127, 131071]	For Y-tiled surfaces	[128B,128KB] = [1 tile, 1024 tiles]
Value	Name	Description												
[63, 131071]	For other linear surfaces	[64B, 128KB]												
[511, 131071]	For X-tiled surface	[512B, 128KB] = [1tile, 256 tiles]												
[127, 131071]	For Y-tiled surfaces	[128B,128KB] = [1 tile, 1024 tiles]												
2	Half Pitch for Chroma Format: _____ Enable This field indicates that the chroma plane(s) will use a pitch equal to half the value specified in the Surface Pitch field. This field is only used for PLANAR surface formats. <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p style="text-align: center; color: blue; font-weight: bold;">Programming Notes</p> <p>Must be programmed to Zero always as this field is not used</p> </div>													
1	Tiled Surface Format: _____ Boolean This field specifies whether the surface is tiled. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 40%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>True</td> <td>Tiled</td> </tr> <tr> <td>0</td> <td>False</td> <td>Linear</td> </tr> </tbody> </table>		Value	Name	Description	1	True	Tiled	0	False	Linear			
Value	Name	Description												
1	True	Tiled												
0	False	Linear												



VEBOX_SURFACE_STATE										
		<p style="text-align: center;">Programming Notes</p> <p>Linear surfaces can be mapped to Main Memory (uncached) or System Memory (cacheable, snooped). Tiled surfaces can only be mapped to Main Memory. The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this bit.</p>								
	0	<p>Tile Walk</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>3D_TileWalk</td> </tr> </table> <p>This field specifies the type of memory tiling (XMajor or YMajor) employed to tile this surface. See <i>Memory Interface Functions</i> for details on memory tiling and restrictions. This field is ignored when the surface is linear.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>TILEWALK_XMAJOR</td> </tr> <tr> <td>1</td> <td>TILEWALK_YMAJOR</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this bit.</p>	Format:	3D_TileWalk	Value	Name	0	TILEWALK_XMAJOR	1	TILEWALK_YMAJOR
Format:	3D_TileWalk									
Value	Name									
0	TILEWALK_XMAJOR									
1	TILEWALK_YMAJOR									
4	31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
	Format:	MBZ								
	28:16	<p>X Offset for U</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>U13 Pixel Offset</td> </tr> </table> <p>This field must be zero for the VEBOX surface formats</p>	Format:	U13 Pixel Offset						
	Format:	U13 Pixel Offset								
15	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
14:0	<p>Y Offset for U</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>U15 Row Offset</td> </tr> </table> <p>This field specifies the vertical offset in rows from the start (origin) or the Luma(Y) plane 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.</p> <p style="text-align: center;">Programming Notes</p> <p>This field must indicate an even number (bit 0 = 0). This field must be evenly divisible by 4 for Tile-Y surfaces (so the offset points to the start of a cache line) For Planar formats, if the surface is in YS or YF tile modes, the Y Offset for U should be an integral multiple of the Tile height of the Luma plane</p>	Format:	U15 Row Offset							
Format:	U15 Row Offset									
5	31:29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ									



VEBOX_SURFACE_STATE			
	28:16 X Offset for V Format: U13 Pixel Offset This field must be zero for the VEBOX surface formats.		
	15 Reserved Format: MBZ		
	14:0 Y Offset for V Format: U15 Row Offset This field specifies the vertical offset in rows from the start (origin) of the Luma(Y) plane to the start (origin) of the V(Cr) plane. This field is only used for PLANAR surface formats with Interleave Chroma disabled. <table border="1" style="margin-top: 10px;"> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> <tr> <td> This field must indicate an even number (bit 0 = 0). This field must be evenly divisible by 4 for Tile-Y surfaces (so the offset points to the start of a cache line). For Planar formats, if the surface is in YS or YF tile modes, the Y Offset for V should be an integral multiple of the Tile height of the Luma plane </td> </tr> </table>	Programming Notes	This field must indicate an even number (bit 0 = 0). This field must be evenly divisible by 4 for Tile-Y surfaces (so the offset points to the start of a cache line). For Planar formats, if the surface is in YS or YF tile modes, the Y Offset for V should be an integral multiple of the Tile height of the Luma plane
	Programming Notes		
This field must indicate an even number (bit 0 = 0). This field must be evenly divisible by 4 for Tile-Y surfaces (so the offset points to the start of a cache line). For Planar formats, if the surface is in YS or YF tile modes, the Y Offset for V should be an integral multiple of the Tile height of the Luma plane			
6			
31 Reserved Format: MBZ			
	30:16 X Offset for Frame Format: U15 Pixel Offset This is an offset in X from the Surface Base Address in pixels for all planes using this surface. For U/V planes this is added to the X Offset for U/V. After converting to bytes this must be an integer multiple of cache lines in the tiling mode. This specifies the edge of the frame, so adjacent pixels needed for Denoise/Deinterlace/Demosaic are replicated or mirrored. <table border="1" style="margin-top: 10px;"> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> <tr> <td> If Y Offset for Frame >0 the X Offset must be 0. If memory compression is enabled then this must be an even number of cache lines. </td> </tr> </table>	Programming Notes	If Y Offset for Frame >0 the X Offset must be 0. If memory compression is enabled then this must be an even number of cache lines.
	Programming Notes		
	If Y Offset for Frame >0 the X Offset must be 0. If memory compression is enabled then this must be an even number of cache lines.		
	15 Reserved Format: MBZ		
14:0 Y Offset for Frame Format: U15 Pixel Offset This is an offset in Y from the Surface Base Address in pixels for all planes using this surface. For U/V planes this is added to the Y Offset for U/V. After converting to bytes this must be an integer multiple of cache lines in the tiling mode. This specifies the edge of the frame, so adjacent pixels needed for Denoise/Deinterlace/Demosaic are replicated or mirrored. <table border="1" style="margin-top: 10px;"> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> <tr> <td> If X Offset for Frame >0 the Y Offset must be 0. For Planar formats, if the surface is in YS or YF tile modes, the Y Offset for Frame should be an integral multiple of the Tile height. </td> </tr> </table>	Programming Notes	If X Offset for Frame >0 the Y Offset must be 0. For Planar formats, if the surface is in YS or YF tile modes, the Y Offset for Frame should be an integral multiple of the Tile height.	
Programming Notes			
If X Offset for Frame >0 the Y Offset must be 0. For Planar formats, if the surface is in YS or YF tile modes, the Y Offset for Frame should be an integral multiple of the Tile height.			



VEBOX_SURFACE_STATE				
7	31:17	Reserved		
	Format:		MBZ	
7	16:0	Derived Surface Pitch		
	Format:		U17 pitch in (Bytes - 1)	
	<p>This field specifies the surface pitch in (#Bytes - 1) for the derived surfaces: STMM/Denoise statistic surface is described when the Surface Identification bit is 0 (Input Surface).</p> <p>The (Current Denoise Output)/(Previous Denoise Input) surfaces are described when the bit is 1 (Output Surface).</p>			
	Value	Name	Description	Exists If
	[63, 131071]		[64B, 128KB]	[Tiled Surface] == 0
	[511, 131071]		[512B, 128KB] = [1tile, 256 tiles]	([Tiled Surface] == 1) AND ([Tile Walk] == 0)
[127, 131071]		[128B,128KB] = [1 tile, 1024 tiles]	([Tiled Surface] == 1) AND ([Tile Walk] == 1)	
Programming Notes				
In DN Only mode, the pitch for the (Current Denoise Output)/(Previous Denoise Input) and the Surface Pitch must be programmed the same.				
The pitch must be a multiple of the tile width.				
8	31:17	Reserved		
	Format:		MBZ	
8	16:0	Surface Pitch for Skin Score Output Surfaces		
	Format:		U17 pitch in (Bytes - 1)	
	<p>This field specifies the surface pitch in (#Bytes - 1) for the Skin Score Output surface if enabled; This is present only in the output surface format and reserved for Input surface format. The height and width are the same as in the Output surface mentioned above.</p>			
	Value	Name	Description	Exists If
	[63, 131071]		[64B, 128KB]	[Tiled Surface] == 0
	[511, 131071]		[512B, 128KB] = [1tile, 256 tiles]	([Tiled Surface] == 1) AND ([Tile Walk] == 0)
[127, 131071]		[128B,128KB] = [1 tile, 1024 tiles]	([Tiled Surface] == 1) AND ([Tile Walk] == 1)	
Programming Notes				
The pitch must be a multiple of the tile width.				



VEBOX_TILING_CONVERT

VEBOX_TILING_CONVERT			
Source:	VideoEnhancementCS		
Length Bias:	2		
This command takes the input surface and writes directly to the output surface at high speed. The surface format and width/height of the input and output must be the same, only the tiling mode and pitch can change.			
DWord	Bit	Description	
0	31:29	Command Type	
		Default Value:	3h PARALLEL_VIDEO_PIPE
		Format:	OpCode
	28:27	Pipeline	
		Default Value:	2h Media
		Format:	OpCode
	26:24	Command OpCode	
		Default Value:	4h VEBOX
		Format:	OpCode
	23:21	SubOpCode A	
Default Value:		0h	
Format:		OpCode	
20:16	SubOpCode B		
	Default Value:	1h	
	Format:	OpCode	
15:12	Reserved		
	Format:	MBZ	
11:0	DWord Length		
	Format:	=n Total Length - 2	
	Value	Name	Description
	3h		(Excludes DWords 0, 1)
1.2	63:12	Input Address	
		Format:	GraphicsAddress63-12
Specifies bits 47:12 of the 4Kbyte-aligned frame buffer address for reading the current frame.			
1.2	11	Reserved	
		Format:	MBZ



VEBOX_TILING_CONVERT		
	10:0	Input Surface Control Bits Format: VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS
3.4	63:12	Output Address Format: GraphicsAddress63-12 Specifies bits 47:12 of the 4Kbyte-aligned frame buffer address for writing the current frame.
	11	Reserved Format: MBZ
	10:0	Output Surface Control Bits Format: VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS



Wait Notification

wait - Wait Notification			
Source:	Eulsa		
Length Bias:	4		
<p>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.</p> <p>When multiple notifications are received, software must use wait instructions to decrement notification count registers for each notification.</p> <p>Notification register n0.0:ud is for thread to thread communication (via the Message Gateway shared function) and n0.1:ud for host to thread communication (through MMIO registers). See the Message Gateway chapter for thread-thread communication for host-to-thread communication.</p>			
Format:	wait (exec_size) nreg		
Restriction			
Restriction : src0 and dst must be n0.0, n0.1, or n0.2.			
Restriction : Execution size must be 1 as the notification registers are scalar.			
Restriction : Predication is not allowed.			
Restriction : Two back-to-back wait instructions are not allowed. At minimum, a nop instruction must be inserted between two wait instructions			
Syntax			
wait (1) n#			
Pseudocode			
N/A			
Predication	Conditional Modifier	Saturation	Source Modifier
N	N	N	N
Src Types	Dst Types		
UD	UD		
DWord	Bit	Description	
0	127:64	Sources	



wait - Wait Notification				
		Exists If:	([[Operand Control]][Src1.RegFile]!='IMM')	
		Format:	EU_INSTRUCTION_SOURCES_REG	
	127:64	Sources		
		Exists If:	([[Operand Control]][Src1.RegFile]='IMM')	
		Format:	EU_INSTRUCTION_SOURCES_IMM32	
	63:32	Operand Control		
		Format:	EU_INSTRUCTION_OPERAND_CONTROLS	
	31:0	Header		
		Format:	EU_INSTRUCTION_HEADER	



While

while - While					
Source:	Eulsa				
Length Bias:	4				
<p>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. 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 ON, none of the PcIP are updated.</p>					
<p>Format: [(pred)] while (exec_size) JIP</p>					
Restriction					
<p>Restriction : The execution size must be the same for the while instruction and any break and cont instructions of the same code block.</p>					
Syntax					
<p>[(pred)] while (exec_size) imm32</p>					
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>					
Predication	Conditional Modifier	Saturation	Source Modifier		
Y	N	N	N		
DWord	Bit	Description			
0..3	127:96	JIP Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>S31</td></tr></table> Jump Target Offset. The relative offset in bytes if a jump is taken for the instruction.			S31
		S31			
	95	Source 0 Address Immediate [9] Sign Bit			
	94:91	Src1.SrcType Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>SrcType</td></tr></table>			SrcType
	SrcType				
90:89	Src1.RegFile Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>RegFile</td></tr></table>			RegFile	
	RegFile				



while - While	
88:64	Source 0
	Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16')
	Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
88:64	Source 0
	Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1')
	Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1
63:32	Operand Control
	Format: EU_INSTRUCTION_OPERAND_CONTROLS
31:0	Header
	Format: EU_INSTRUCTION_HEADER



XY_COLOR_BLT

XY_COLOR_BLT			
Source:	BlitterCS		
Length Bias:	2		
<p>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.</p> <p>This instruction is optimized to run at the maximum memory write bandwidth.</p> <p>The typical (and fastest) Raster operation code = F0 which performs a copy of the pattern background register to the destination.</p>			
DWord	Bit	Description	
0 BR00	31:29	Client	
		Default Value:	02h 2D Processor
		Format:	Opcode
	28:22	Instruction Target(Opcode)	
		Default Value:	50h
		Format:	Opcode
	21:20	32bpp Byte Mask	
		This field is only used for 32bpp.	
		Value	Name
		1xb	Write Alpha Channel
x1b	Write RGB Channel		
19:12	Reserved		
	Format:	MBZ	
11	Tiling Enable		
	Value	Name	
	0b	Tiling Disabled (Linear Blit)	
	1b	Tiling Enabled	Tile-X or Tile-Y
10:8	Reserved		
	Format:	MBZ	
7:0	DWord Length		
	Default Value:	05h	
	Format:	=n	
1 BR13	31	Reserved	
		Format:	MBZ



XY_COLOR_BLT			
	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 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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.	
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.	
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.	
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.	
4..5	63:0	Destination Base Address Format: GraphicsAddress63-0 This address must be Cache Line (64Byte) aligned for all surface types (linear or tiled).	
6 BR16	31:0	Solid Pattern Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]	



XY_FAST_COPY_BLT

XY_FAST_COPY_BLT				
Source:	BlitterCS			
Length Bias:	2			
<p>This BLT instruction performs a color source copy where the only operands involved are a color source and destination of the same bit width. The source and destination surfaces CANNOT overlap. The hardware assumes this whenever this Fast_Copy command is given to it. For overlapping Blits, use the traditional XY_SRC_COPY_BLT command (for overlap determination, read the description given in the XY_SRC_COPY_BLT command). Note that this command does not support Clipping operations. This new blit command will happen in large numbers, consecutively, possibly an entire batch will comprise only new blit commands Legacy commands and new blit command will not be interspersed. If they are, they will be separated by implied HW flush: Whenever there is a transition between this new Fast Blit command and the Legacy Blit commands, the HW will impose an automatic flush BEFORE the execution (at the beginning) of the next blitter command. New blit command can use any combination of memory surface type - linear, tiledX, tiledY, and the tiling information is conveyed as part of the new Fast Copy command. The Fast Copy Blit supports the new 64KB Tiling defined. The starting pixel of Fast Copy blit for both source and destination should be on an OWord boundary.</p> <p>Note that when two sequential fast copy blits have different source surfaces, but their destinations refer to the same destination surfaces and therefore destinations overlap it is imperative that a Flush be inserted between the two blits.</p>				
DWord	Bit	Description		
0 BR00	31:29	Client		
		Default Value:	02h 2D Processor	
		Format:	Opcode	
	28:22	Instruction Target(Opcode)		
		Default Value:	42h	
		Format:	Opcode	
	21:20	Source Tiling Method		
		SW is required to flush the HW before changing the polarity of these bits for subsequent blits.		
		Value	Name	Description
		00b	Linear (Tiling Disabled)	
01b		Legacy Tile-X		
10b		Tile-Y	Choosing between 'Legacy Tile-Y' or the 'New 4K Tile-YF' can be done in DWord 1, Bit[31].	
11b	64kb Tiling			
19:15	Reserved			
	Format:	MBZ		



XY_FAST_COPY_BLT

	14:13	Destination Tiling Method SW is required to flush the HW before changing the polarity of these bits for subsequent blits.	
		Value	Name
		00b	Linear (Tiling Disabled)
		01b	Legacy Tile-X
		10b	Tile-Y
			Choosing between 'Legacy Tile-Y' or the 'New 4K Tile-YF' can be done in DWord 1, Bit[30].
		11b	64kb Tiling
	12:8	Reserved Format: MBZ	
	7:0	DWord Length Default Value: 08h Excludes DWORD 0,1 Format: =n 08h	
	1 BR13	31	Tile Y Type for Source Source being Tile-Y can be selected in DWord 0, Bit[21:20].
		Value	Name
		0b	Legacy Tile-Y
		1b	New 4k Tile-YF
30		Tile Y Type for Destination Destination being Tile-Y can be selected in DWord 0, Bit[14:13].	
		Value	Name
		0b	Legacy Tile-Y
		1b	New 4k Tile-YF
29:27		Reserved Format: MBZ	
26:24		Color Depth	
	Value	Name	
	000b	8 bit color	
	001b	16 bit color (565)	
	011b	32 bit color	
	100b	64 bit color (for 64KB Tiling)	
	101b	128 bit color (for 64KB Tiling)	
23:16	Reserved Format: MBZ		



XY_FAST_COPY_BLT				
	15:0	<p>Destination Pitch (signed) 2's Complement. For Fast Copy Blits the pitch cannot be a negative number. As a result the pitch must be a positive value and bit 15 of the destination pitch must be zero. For Linear surfaces, the pitch has to be an OWord (16byte) multiple for Linear-Linear copies; and CacheLine aligned for Tiled-Linear type copies. The value should be specified as a number in <i>bytes</i>. For Tiled surfaces, the pitch has to be a multiple of the Tile width (X direction width of the Tile). This means the pitch value will always be Cache Line aligned (64byte multiple) and the number or value mentioned in this field here should be specified as a number of Dwords (4byte quantity).</p>		
2 BR22	31:16	<p>Destination Y1 Coordinate (Top) 16-bit signed number.</p>		
	15:0	<p>Destination X1 Coordinate (Left) 16-bit signed number. The start pixel for Fast Copy blit should be on an OWord boundary.</p>		
3 BR23	31:16	<p>Destination Y2 Coordinate (Bottom) 16-bit signed number.</p>		
	15:0	<p>Destination X2 Coordinate (Right) 16-bit signed number.</p>		
4..5	63:0	<p>Destination Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table> <p>This address must be Cache Line (64Byte) aligned for all surface types (linear or tiled).</p>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
6 BR26	31:16	<p>Source Y1 Coordinate (Top) 16-bit signed number.</p>		
	15:0	<p>Source X1 Coordinate (Left) 16-bit signed number. The start pixel for Fast Copy blit should be on an OWord boundary.</p>		
7 BR11	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
15:0	<p>Source Pitch (double word aligned) 2's Complement. For Fast Copy Blits the pitch cannot be a negative number. As a result the pitch must be a positive value and bit 15 of the source pitch must be zero. For Linear surfaces, the pitch has to be an OWord (16byte) multiple for Linear-Linear copies; and CacheLine aligned for Linear-Tiled type copies. The value , and the value should be specified as a number in <i>bytes</i>. For Tiled surfaces, the pitch has to be a multiple of the Tile width (X direction width of the Tile). This means the pitch value will always be Cache Line aligned (64byte multiple) and the number or value mentioned in this field here should be specified as a number of Dwords (4byte quantity).</p>			
8..9	63:0	<p>Source Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table> <p>This address must be Cache Line (64Byte) aligned for all surface types (linear or tiled).</p>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			



XY_FULL_BLT

XY_FULL_BLT											
Source:	BlitterCS										
Length Bias:	2										
<p>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.</p> <p>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.</p> <p>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.</p>											
DWord	Bit	Description									
0 BR00	31:29	Client									
		Default Value:	02h 2D Processor								
		Format:	Opcode								
	28:22	Instruction Target(Opcode)									
		Default Value:	55h								
Format:		Opcode									
21:20	32bpp Byte Mask This field is only used for 32bpp.	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>[Default]</td> </tr> <tr> <td>1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td>x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	00b	[Default]	1xb	Write Alpha Channel	x1b	Write RGB Channel	
		Value	Name								
		00b	[Default]								
		1xb	Write Alpha Channel								
		x1b	Write RGB Channel								
19:16	Reserved										
	Format:	MBZ									
15	Src Tiling Enable	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
		Value	Name	Description							
		0b	Tiling Disabled (Linear Blit)								
		1b	Tiling Enabled	: Tile-X or Tile-Y.							



XY_FULL_BLT			
	14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.	
	11	Dest Tiling Enable	
		Value	Name
		0b	Tiling Disabled (Linear Blit)
	1b	Tiling Enabled	: 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 Length		
	Default Value:	0Ah	
1 BR13	31	Reserved	
		Format:	MBZ
	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 DWords 2's complement For Tiled surfaces (bit_15 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).		
2 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.	
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.	
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.	
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.	



XY_FULL_BLT				
4..5	63:0	<p>Destination Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table> <p>Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address must be CL (64byte) aligned.</p>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
6 BR11	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table> <p>Should be programmed all 0's for 48bit addressing.</p>	Format:	MBZ
	Format:	MBZ		
15:0	<p>Source Pitch (double word aligned and signed) and in DWords</p> <p>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).</p>			
7 BR26	31:16	<p>Source Y1 Coordinate (Top)</p> <p>16 bit signed number.</p>		
	15:0	<p>Source X1 Coordinate (Left)</p> <p>16 bit signed number.</p>		
8..9	63:0	<p>Source Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table> <p>Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address must be CL (64byte) aligned.</p>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
10..11	63:0	<p>Pattern Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table> <p>Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address must be CL (64byte) aligned.</p>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			



XY_FULL_IMMEDIATE_PATTERN_BLT

XY_FULL_IMMEDIATE_PATTERN_BLT			
Source:	BlitterCS		
Length Bias:	2		
<p>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.</p> <p>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.</p> <p>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.</p>			
DWord	Bit	Description	
0 BR00	31:29	Client	
		Default Value:	02h 2D Processor
		Format:	Opcode
	28:22	Instruction Target(Opcode)	
		Default Value:	74h
		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	
19:16	Reserved		
	Format:	MBZ	
15	Src Tiling Enable		
	Value	Name	Description
	0b	Tiling Disabled (Linear)	
	1b	Tiling Enabled	: Tile-X or Tile-Y.



XY_FULL_IMMEDIATE_PATTERN_BLT			
	14:12	Pattern Horizontal Seed (pixel of the scan line to start on corresponding to DST X=0)	
	11	Dest Tiling Enable	
		Value	Name
		0b	Tiling Disabled (Linear Blit)
	1b	Tiling Enabled	: 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 Length		
	Default Value:	08h Excludes DWORD 0,1	
08 + DWL = (Number of Immediate double words)h			
1 BR13	31	Reserved	
		Format:	MBZ
	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 DWords 2's complement For Tiled surfaces (bit_15 enabled) this pitch is of 512Byte granularity for Tile-X, 128B granularity for Tile-Y and can be upto 128Kbytes (or 32KDwords).		
2 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.	
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.	
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.	
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.	



XY_FULL_IMMEDIATE_PATTERN_BLT		
4..5	63:0	Destination Base Address Format: GraphicsAddress63-0 Base address of the destination surface: X=0, Y=0. When Src Tiling is enabled (Bit_15 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address should be CL (64byte) aligned.
		Reserved Format: MBZ Should be programmed all 0's for 48bit addressing.
6 BR11	31:16	Reserved Format: MBZ 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).
7 BR26	31:16	Source Y1 Coordinate (Top) 16 bit signed number.
	15:0	Source X1 Coordinate (Left) 16 bit signed number.
8..9	63:0	Source Address Format: GraphicsAddress63-0 Base address of the source surface: X=0, Y=0. When Src Tiling is enabled (Bit_15 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address should be CL (64byte) aligned.
		Immediate Data 0
10..n	31:0	Immediate Data 0



XY_FULL_MONO_PATTERN_BLT

XY_FULL_MONO_PATTERN_BLT			
Source:	BlitterCS		
Length Bias:	2		
<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>			
DWord	Bit	Description	
0 BR00	31:29	Client	
		Default Value:	02h 2D Processor
		Format:	Opcode
	28:22	Instruction Target(Opcode)	
		Default Value:	57h
		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		
19:16	Reserved		



XY_FULL_MONO_PATTERN_BLT											
	15	Src Tiling Enable <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
	Value	Name	Description								
	0b	Tiling Disabled (Linear Blit)									
	1b	Tiling Enabled	: Tile-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 <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
	Value	Name	Description								
	0b	Tiling Disabled (Linear Blit)									
	1b	Tiling Enabled	: 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 Length <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0Ch</td> <td></td> </tr> </tbody> </table>	Value	Name	0Ch							
Value	Name										
0Ch											
1 BR13	31	Solid Pattern Select <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No Solid Pattern</td> </tr> <tr> <td>1</td> <td>Solid Pattern</td> </tr> </tbody> </table>	Value	Name	0	No Solid Pattern	1	Solid Pattern			
	Value	Name									
	0	No Solid Pattern									
	1	Solid Pattern									
	30	Clipping Enabled <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
	29	Reserved Format: MBZ									
	28:27	Mono Source Transparency Mode <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Use Background</td> </tr> <tr> <td>1</td> <td>Transparency Enabled</td> </tr> </tbody> </table>	Value	Name	0	Use Background	1	Transparency Enabled			
Value	Name										
0	Use Background										
1	Transparency Enabled										
26	Reserved Format: MBZ										
25:24	Color Depth <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)		
Value	Name										
00b	8 Bit Color										
01b	16 Bit Color(565)										
10b	16 Bit Color(1555)										



XY_FULL_MONO_PATTERN_BLT				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">11b</td> <td>32 Bit Color</td> </tr> </table>	11b	32 Bit Color
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.		
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.		
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.		
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.		
4..5	63:0	Destination Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table> Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_15 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address should be CL (64byte) aligned.	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
6 BR11	31:16	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
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).			
7 BR26	31:16	Source Y1 Coordinate (Top) 16 bit signed number.		
	15:0	Source X1 Coordinate (Left) 16 bit signed number.		
8..9	63:0	Source Address <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table> Base address of the source surface: X=0, Y=0. When Src Tiling is enabled (Bit_15 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address should be CL (64byte) aligned.	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
10 BR16	31:0	Pattern Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]		
11 BR17	31:0	Pattern Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]		
12 BR20	31:0	Pattern Data 0 (least significant DW)		



XY_FULL_MONO_PATTERN_BLT		
13 BR21	31:0	Pattern Data 1 (most significant DW)



XY_FULL_MONO_PATTERN_MONO_SRC_BLT

XY_FULL_MONO_PATTERN_MONO_SRC_BLT									
Source:	BlitterCS								
Length Bias:	2								
<p>The full BLT provides the ability to specify all 3 operands: destination, source, and pattern. The pattern and source operands are monochrome.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Negative Stride (= Pitch) is NOT ALLOWED.</p>									
DWord	Bit	Description							
0 BR00	31:29	Client <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode			
	Default Value:	02h 2D Processor							
	Format:	Opcode							
	28:22	Instruction Target(Opcode) <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>58h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	58h	Format:	Opcode			
Default Value:	58h								
Format:	Opcode								
21:20	32bpp Byte Mask This field is only used for 32bpp. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td style="text-align: center;">[Default]</td> </tr> <tr> <td style="text-align: center;">1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td style="text-align: center;">x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	00b	[Default]	1xb	Write Alpha Channel	x1b	Write RGB Channel
Value	Name								
00b	[Default]								
1xb	Write Alpha Channel								
x1b	Write RGB Channel								
19:17	Monochrome source data bit position of the first pixel within a byte per scan line.								



XY_FULL_MONO_PATTERN_MONO_SRC_BLT										
	16:15 Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ								
	14:12 Pattern Horizontal Seed (pixel of the scan line to start on corresponding to DST X=0)									
	11 Tiling Enable <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 55%;">Name</th> <th style="width: 30%;">Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
	Value	Name	Description							
0b	Tiling Disabled (Linear Blit)									
1b	Tiling Enabled	: 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 Length <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0Ch</td> <td></td> </tr> </tbody> </table>	Value	Name	0Ch							
Value	Name									
0Ch										
1 BR13	31 Solid Pattern Select <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No Solid Pattern</td> </tr> <tr> <td>1</td> <td>Solid Pattern</td> </tr> </tbody> </table>	Value	Name	0	No Solid Pattern	1	Solid Pattern			
	Value	Name								
	0	No Solid Pattern								
	1	Solid Pattern								
	30 Clipping Enabled <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name								
	0b	Disabled								
1b	Enabled									
29 Mono Source Transparency Mode <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Use Background</td> </tr> <tr> <td>1</td> <td>Transparency Enabled</td> </tr> </tbody> </table>	Value	Name	0	Use Background	1	Transparency Enabled				
Value	Name									
0	Use Background									
1	Transparency Enabled									
28 Mono Pattern Transparency Mode <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Use Background</td> </tr> <tr> <td>1</td> <td>Transparency Enabled</td> </tr> </tbody> </table>	Value	Name	0	Use Background	1	Transparency Enabled				
Value	Name									
0	Use Background									
1	Transparency Enabled									
27:26 Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ									
25:24 Color Depth <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color						
Value	Name									
00b	8 Bit Color									



XY_FULL_MONO_PATTERN_MONO_SRC_BLT								
		<table border="1"> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </table>	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.						
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.						
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.						
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.						
4..5 This bitfield contains the base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Destination Base Address <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0				
Format:	GraphicsAddress63-0							
6..7 Address corresponds to DST X1, Y1. Note no NPO2 change here. The Mono Source Base Address must always be Cache Line (64byte) aligned.	63:0	Mono Source Address <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0				
Format:	GraphicsAddress63-0							
8 BR18	31:0	Source Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]						
9 BR19	31:0	Source Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]						
10 BR16	31:0	Pattern Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]						
11 BR17	31:0	Pattern Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]						
12 BR20	31:0	Pattern Data 0 (least significant DW)						
13 BR21	31:0	Pattern Data 1 (most significant DW)						



XY_FULL_MONO_SRC_BLT

XY_FULL_MONO_SRC_BLT			
Source:	BlitterCS		
Length Bias:	2		
<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Negative Stride (= Pitch) is NOT ALLOWED</p>			
DWord	Bit	Description	
0 BR00	31:29	Client	
		Default Value: 02h 2D Processor	
		Format: Opcode	
	28:22	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 Channel		
19:17	Monochrome source data bit position of the first pixel within a byte per scan line.		
16:15	Reserved		
	Format: MBZ		



XY_FULL_MONO_SRC_BLT											
	14:12	Pattern Horizontal Seed (pixel of the scan line to start on corresponding to DST X=0)									
	11	Tiling Enable									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
		Value	Name	Description							
0b	Tiling Disabled (Linear Blit)										
1b	Tiling Enabled	: 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 Length										
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0Ah</td> <td></td> </tr> </tbody> </table>	Value	Name	0Ah							
Value	Name										
0Ah											
1 BR13	31	Reserved Format: _____ MBZ									
	30	Clipping Enabled									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
		Value	Name								
	0b	Disabled									
	1b	Enabled									
	29	Mono Source Transparency Mode									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Use Background</td> </tr> <tr> <td>1</td> <td>Transparency Enabled</td> </tr> </tbody> </table>	Value	Name	0	Use Background	1	Transparency Enabled			
		Value	Name								
	0	Use Background									
1	Transparency Enabled										
28:26	Reserved Format: _____ MBZ										
25:24	Color Depth										
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
	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).										



XY_FULL_MONO_SRC_BLT		
2 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.
4..5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address must be CL (64byte) aligned.	63:0	Destination Base Address Format: GraphicsAddress63-0
6..7 Address corresponds to DST X1, Y1. Note no NPO2 change here. The Mono Source Base Address must always be Cache Line (64byte) aligned.	63:0	Mono Source Address Format: GraphicsAddress63-0
8 BR18	31:0	Source Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
9 BR19	31:0	Source Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
10..11 28:06 are implemented. Note no NPO2 change here. The pattern data must be located in linear memory. The programmed Pattern Base Address must always be Cache Line (64byte) aligned.	63:0	Pattern Base Address Format: GraphicsAddress63-0



XY_FULL_MONO_SRC_IMMEDIATE_PATTERN_BLT

XY_FULL_MONO_SRC_IMMEDIATE_PATTERN_BLT										
Source:	BlitterCS									
Length Bias:	2									
<p>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.</p>										
DWord	Bit	Description								
0 BR00	31:29	Client <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode				
	Default Value:	02h 2D Processor								
	Format:	Opcode								
	28:22	Instruction Target(Opcode) <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>75h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	75h	Format:	Opcode				
	Default Value:	75h								
	Format:	Opcode								
	21:20	32bpp Byte Mask This field is only used for 32bpp. <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td style="text-align: center;">[Default]</td> </tr> <tr> <td style="text-align: center;">1xb</td> <td style="text-align: center;">Write Alpha Channel</td> </tr> <tr> <td style="text-align: center;">x1b</td> <td style="text-align: center;">Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	00b	[Default]	1xb	Write Alpha Channel	x1b	Write RGB Channel
	Value	Name								
	00b	[Default]								
	1xb	Write Alpha Channel								
x1b	Write RGB Channel									
19:17	Monochrome source data bit position of the first pixel within a byte per scan line.									
16:15	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
14:12	Pattern Horizontal Seed (pixel of the scan line to start on corresponding to DST X=0)									



XY_FULL_MONO_SRC_IMMEDIATE_PATTERN_BLT

	11	Tiling Enable	
		Value	Name
	0b	Tiling Disabled (Linear Blit)	
	1b	Tiling Enabled : 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 Length	
		Default Value:	08h Excludes DWORD 0,1
		08 + DWL = (Number of Immediate double words)h	
1 BR13	31	Reserved	
		Format:	MBZ
	30	Clipping Enabled	
		Value	Name
		0b	Disabled
		1b	Enabled
	29	Mono Source Transparency 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(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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.	



XY_FULL_MONO_SRC_IMMEDIATE_PATTERN_BLT		
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.
4..5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Destination Base Address
		Format: GraphicsAddress63-0
6..7 Address corresponds to DST X1, Y1. Note no NPO2 change here. The Mono Source Base Address must always be Cache Line (64byte) aligned.	63:0	Mono Source Address
		Format: GraphicsAddress63-0
8 BR18	31:0	Source Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
9 BR19	31:0	Source Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]
10..n	31:0	Immediate Data



XY_MONO_PAT_BLT

XY_MONO_PAT_BLT										
Source:	BlitterCS									
Length Bias:	2									
<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>										
DWord	Bit	Description								
0 BR00	31:29	Client <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode				
	Default Value:	02h 2D Processor								
	Format:	Opcode								
	28:22	Instruction Target(Opcode) <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>52h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	52h	Format:	Opcode				
	Default Value:	52h								
	Format:	Opcode								
	21:20	32bpp Byte Mask This field is only used for 32bpp. <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td style="text-align: center;">[Default]</td> </tr> <tr> <td style="text-align: center;">1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td style="text-align: center;">x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	00b	[Default]	1xb	Write Alpha Channel	x1b	Write RGB Channel
Value	Name									
00b	[Default]									
1xb	Write Alpha Channel									
x1b	Write RGB Channel									
19:15	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.									
11	Tiling Enable <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
Value	Name	Description								
0b	Tiling Disabled (Linear Blit)									
1b	Tiling Enabled	: Tile-X or Tile-Y.								



XY_MONO_PAT_BLT											
	10:8	Pattern Vertical Seed Scan line of the 8x8 pattern to start on corresponding to DST Y=0.									
	7:0	DWord Length <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>08h</td> <td></td> </tr> </tbody> </table>	Value	Name	08h						
Value	Name										
08h											
1 BR13	31	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	30	Clipping Enabled <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
	29	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ										
28	Mono Pattern Transparency Mode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Use Background</td> </tr> <tr> <td>1</td> <td>Transparency Enabled</td> </tr> </tbody> </table>	Value	Name	0	Use Background	1	Transparency Enabled				
Value	Name										
0	Use Background										
1	Transparency Enabled										
27:26	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
25:24	Color Depth <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.									
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.									
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.									



XY_MONO_PAT_BLT						
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.				
4..5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	<table border="1" style="width: 100%;"> <tr> <td colspan="2">Destination Base Address</td> </tr> <tr> <td style="width: 150px;">Format:</td> <td style="text-align: center;">GraphicsAddress63-0</td> </tr> </table>	Destination Base Address		Format:	GraphicsAddress63-0
Destination Base Address						
Format:	GraphicsAddress63-0					
6 BR16	31:0	Pattern Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]				
7 BR17	31:0	Pattern Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]				
8 BR20	31:0	Pattern Data 0				
9 BR21	31:0	Pattern Data 1				



XY_MONO_PAT_FIXED_BLT

XY_MONO_PAT_FIXED_BLT			
Source:	BlitterCS		
Length Bias:	2		
<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>			
DWord	Bit	Description	
0 BR00	31:29	Client	
		Default Value:	02h 2D Processor
		Format:	Opcode
	28:22	Instruction Target(Opcode)	
		Default Value:	59h
		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	Reserved		
	Format:	MBZ	
18:15	Fixed Pattern		
	Value	Name	
	0000b	HS_HORIZONTAL	
	0001b	HS_VERTICAL	
	0010b	HS_FDIAGONAL	
0011b	HS_BDIAGONAL		



XY_MONO_PAT_FIXED_BLT																										
		<table border="1"> <tr><td>0100b</td><td>HS_CROSS</td></tr> <tr><td>0101b</td><td>HS_DIAGCROSS</td></tr> <tr><td>0110b</td><td>Reserved</td></tr> <tr><td>0111b</td><td>Reserved</td></tr> <tr><td>1000b</td><td>Screen Door</td></tr> <tr><td>1001b</td><td>SD Wide</td></tr> <tr><td>1010b</td><td>Walking Bit (one)</td></tr> <tr><td>1011b</td><td>Walking Zero</td></tr> <tr><td>1100b</td><td>Reserved</td></tr> <tr><td>1101b</td><td>Reserved</td></tr> <tr><td>1110b</td><td>Reserved</td></tr> <tr><td>1111b</td><td>Reserved</td></tr> </table>	0100b	HS_CROSS	0101b	HS_DIAGCROSS	0110b	Reserved	0111b	Reserved	1000b	Screen Door	1001b	SD Wide	1010b	Walking Bit (one)	1011b	Walking Zero	1100b	Reserved	1101b	Reserved	1110b	Reserved	1111b	Reserved
0100b	HS_CROSS																									
0101b	HS_DIAGCROSS																									
0110b	Reserved																									
0111b	Reserved																									
1000b	Screen Door																									
1001b	SD Wide																									
1010b	Walking Bit (one)																									
1011b	Walking Zero																									
1100b	Reserved																									
1101b	Reserved																									
1110b	Reserved																									
1111b	Reserved																									
	14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.																								
	11	Tiling Enable <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.															
Value	Name	Description																								
0b	Tiling Disabled (Linear Blit)																									
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 Length <table border="1"> <tr> <td>Format:</td> <td>=n</td> </tr> <tr> <th>Value</th> <th>Name</th> </tr> <tr> <td>06h</td> <td></td> </tr> </table>	Format:	=n	Value	Name	06h																			
Format:	=n																									
Value	Name																									
06h																										
1 BR13	31	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																						
	Format:	MBZ																								
	30	Clipping Enabled <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled																		
Value	Name																									
0b	Disabled																									
1b	Enabled																									
29	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																							
Format:	MBZ																									



XY_MONO_PAT_FIXED_BLT												
	28	Mono Pattern Transparency Mode <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Use Background</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Transparency Enabled</td> </tr> </tbody> </table>	Value	Name	0	Use Background	1	Transparency Enabled				
	Value	Name										
	0	Use Background										
	1	Transparency Enabled										
	27:26	Reserved Format: MBZ										
	25:24	Color Depth <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>8 Bit Color</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
	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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.										
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.										
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.										
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.										
4..5	63:0 Destination Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="color: red;">GraphicsAddress63-0</td> </tr> </table> Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address should be CL (64byte) aligned.	Format:	GraphicsAddress63-0									
Format:	GraphicsAddress63-0											
6 BR16	31:0	Pattern Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]										
7 BR17	31:0	Pattern Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]										



XY_MONO_SRC_COPY_BLT

XY_MONO_SRC_COPY_BLT										
Source:	BlitterCS									
Length Bias:	2									
<p>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.</p> <p>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.</p> <p>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.</p>										
DWord	Bit	Description								
0 BR00	31:29	Client <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode				
	Default Value:	02h 2D Processor								
	Format:	Opcode								
	28:22	Instruction Target(Opcode) <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>54h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	54h	Format:	Opcode				
	Default Value:	54h								
	Format:	Opcode								
	21:20	32bpp Byte Mask This field is only used for 32bpp. <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td style="text-align: center;">[Default]</td> </tr> <tr> <td style="text-align: center;">1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td style="text-align: center;">x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	00b	[Default]	1xb	Write Alpha Channel	x1b	Write RGB Channel
	Value	Name								
	00b	[Default]								
	1xb	Write Alpha Channel								
x1b	Write RGB Channel									
19:17	Monochrome source data bit position of the first pixel within a byte per scan line.									
16:12	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
11	Tiling Enable <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
Value	Name	Description								
0b	Tiling Disabled (Linear Blit)									
1b	Tiling Enabled	: Tile-X or Tile-Y.								



XY_MONO_SRC_COPY_BLT											
	10:8	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
7:0	DWord Length <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>08h</td> <td></td> </tr> </tbody> </table>	Value	Name	08h							
Value	Name										
08h											
1 BR13	31	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	30	Clipping Enabled <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
	29	Mono Source Transparency Mode <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Use Background</td> </tr> <tr> <td>1</td> <td>Transparency Enabled</td> </tr> </tbody> </table>	Value	Name	0	Use Background	1	Transparency Enabled			
Value	Name										
0	Use Background										
1	Transparency Enabled										
28:26	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
25:24	Color Depth <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.									
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.									
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.									
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.									



XY_MONO_SRC_COPY_BLT				
4..5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Destination Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
6..7 Address corresponds to DST X1, Y1. Note no NPO2 change here. The Mono Source Base Address must always be Cache Line (64byte) aligned.	63:0	Mono Source Address <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
8 BR18	31:0	Source Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]		
9 BR19	31:0	Source Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]		



XY_MONO_SRC_COPY_IMMEDIATE_BLT

XY_MONO_SRC_COPY_IMMEDIATE_BLT			
Source:	BlitterCS		
Length Bias:	2		
<p>This instruction allows the Driver to send monochrome data through the instruction stream, eliminating the read latency of the source during command execution.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Negative Stride (= Pitch) is NOT ALLOWED.</p>			
DWord	Bit	Description	
0 BR00	31:29	Client	
		Default Value: 02h 2D Processor	
	Format: Opcode		
	28:22	Instruction Target(Opcode)	
		Default Value: 71h	
	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	Monochrome source data bit position of the first pixel within a byte per scan line.		
16:12	Reserved		
	Format: MBZ		
11	Tiling Enable		
	Value	Name	Description
	0b	Tiling Disabled (Linear)	
	1b	Tiling Enabled	: Tile-X or Tile-Y.



XY_MONO_SRC_COPY_IMMEDIATE_BLT											
	10:8	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Default Value:</td> <td>06h Excludes DWORD 0,1</td> </tr> </table> 06 + DWL = (Number of Immediate double words)h	Default Value:	06h Excludes DWORD 0,1								
Default Value:	06h Excludes DWORD 0,1										
1 BR13	31	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	30	Clipping Enabled <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
	29	Mono Source Transparency Mode <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Transparency Enabled</td> </tr> <tr> <td>1b</td> <td>Use Background</td> </tr> </tbody> </table>	Value	Name	0b	Transparency Enabled	1b	Use Background			
Value	Name										
0b	Transparency Enabled										
1b	Use Background										
28:26	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
25:24	Color Depth <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.									
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.									
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.									
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.									



XY_MONO_SRC_COPY_IMMEDIATE_BLT				
4..5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Destination Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
6 BR18	31:0	Source Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]		
7 BR19	31:0	Source Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0]		
8..n	31:0	Immediate Data		



XY_PAT_BLT

XY_PAT_BLT										
Source:	BlitterCS									
Length Bias:	2									
<p>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.</p>										
DWord	Bit	Description								
0 BR00	31:29	Client <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode				
	Default Value:	02h 2D Processor								
	Format:	Opcode								
	28:22	Instruction Target(Opcode) <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>51h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	51h	Format:	Opcode				
	Default Value:	51h								
	Format:	Opcode								
	21:20	32bpp Byte Mask This field is only used for 32bpp. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>[Default]</td> </tr> <tr> <td style="text-align: center;">1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td style="text-align: center;">x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	00b	[Default]	1xb	Write Alpha Channel	x1b	Write RGB Channel
	Value	Name								
	00b	[Default]								
	1xb	Write Alpha Channel								
x1b	Write RGB Channel									
19:15	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.									
11	Tiling Enable <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
Value	Name	Description								
0b	Tiling Disabled (Linear Blit)									
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 Length <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>06h</td> </tr> </table>	Default Value:	06h							
Default Value:	06h									



XY_PAT_BLT											
1 BR13	31	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	30	Clipping Enabled <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
29:26	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
25:24	Color Depth <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.									
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.									
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.									
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.									
4..5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Destination Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0							
Format:	GraphicsAddress63-0										



XY_PAT_BLT			
6..7	63:0 Pattern Base Address		
28:06 are implemented. Note no NPO2 change here. The pattern data must be located in linear memory. The programmed Pattern Base Address must always be Cache Line (64byte) aligned.	<table border="1" style="width: 100%;"><tr><td style="width: 20%;">Format:</td><td>GraphicsAddress63-0</td></tr></table>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0		



XY_PAT_BLT_IMMEDIATE

XY_PAT_BLT_IMMEDIATE										
Source:	BlitterCS									
Length Bias:	2									
<p>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.</p> <p>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.</p> <p>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.</p>										
DWord	Bit	Description								
0 BR00	31:29	Client <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode				
	Default Value:	02h 2D Processor								
	Format:	Opcode								
	28:22	Instruction Target(Opcode) <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>72h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	72h	Format:	Opcode				
	Default Value:	72h								
	Format:	Opcode								
	21:20	32bpp Byte Mask This field is only used for 32bpp. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>[Default]</td> </tr> <tr> <td style="text-align: center;">1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td style="text-align: center;">x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	00b	[Default]	1xb	Write Alpha Channel	x1b	Write RGB Channel
	Value	Name								
	00b	[Default]								
	1xb	Write Alpha Channel								
x1b	Write RGB Channel									
19:15	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.									
11	Tiling Enable <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
Value	Name	Description								
0b	Tiling Disabled (Linear Blit)									
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 Length									



XY_PAT_BLT_IMMEDIATE											
		<table border="1"> <tr> <td>Default Value:</td> <td>04h Excludes DWORD 0,1</td> </tr> <tr> <td colspan="2">04 + DWL = (Number of Immediate double)h</td> </tr> </table>	Default Value:	04h Excludes DWORD 0,1	04 + DWL = (Number of Immediate double)h						
Default Value:	04h Excludes DWORD 0,1										
04 + DWL = (Number of Immediate double)h											
1 BR13	31	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	30	Clipping Enabled <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
1b	Enabled										
29:26	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
25:24	Color Depth <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.									
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.									
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.									
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.									
4.5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Destination Base Address <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0							
Format:	GraphicsAddress63-0										
6..n	31:0	Immediate Data									



XY_PAT_CHROMA_BLT

XY_PAT_CHROMA_BLT			
Source:	BlitterCS		
Length Bias:	2		
<p>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.</p>			
DWord	Bit	Description	
0 BR00	31:29	Client	
		Default Value:	02h 2D Processor
		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 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 Length		
	Default Value:	08h Excludes DWORD 0,1	



XY_PAT_CHROMA_BLT											
1 BR13	31	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	30	Clipping Enabled <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
29:26	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
25:24	Color Depth <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 40%;">Value</th> <th style="width: 60%;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
Value	Name										
00b	8 Bit Color										
01b	16 Bit Color(565)										
10b	16 Bit Color(1555)										
11b	32 Bit Color										
23:16	Raster Operation <table border="1" style="width: 100%;"> <tr> <td style="width: 10%;">15:0</td> <td> 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). </td> </tr> </table>	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).								
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.									
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.									
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.									
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.									
4.5	63:0	Destination Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table> This address must be Cache Line (64Byte) aligned for all surface types (linear or tiled).	Format:	GraphicsAddress63-0							
Format:	GraphicsAddress63-0										
6..7	63:0	Pattern Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table> (28:06 are implemented) (Note no NPO2 change here). The pattern data must be located in linear memory. The Pattern Base Address must always be Cache Line (64byte) aligned.	Format:	GraphicsAddress63-0							
Format:	GraphicsAddress63-0										
8 BR18	31:0	Transparency Color Low (Chroma-key Low = Pixel Greater or Equal)									
9 BR19	31:0	Transparency Color High (Chroma-key High = Pixel Less or Equal)									



XY_PAT_CHROMA_BLT_IMMEDIATE

XY_PAT_CHROMA_BLT_IMMEDIATE			
Source:	BlitterCS		
Length Bias:	2		
<p>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.</p> <p>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.</p> <p>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.</p>			
DWord	Bit	Description	
0 BR00	31:29	Client	
		Default Value:	02h 2D Processor
		Format:	Opcode
	28:22	Instruction Target(Opcode)	
		Default Value:	77h
		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 Enabled	: Tile-X or Tile-Y.



XY_PAT_CHROMA_BLT_IMMEDIATE											
	10:8	Pattern Vertical Seed Scan line of the 8x8 pattern to start on corresponding to DST Y=0.									
	7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>06h Excludes DWORD 0,1</td> </tr> </table> 06 + DWL = (Number of Immediate double)h	Default Value:	06h Excludes DWORD 0,1							
Default Value:	06h Excludes DWORD 0,1										
1 BR13	31	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	30	Clipping Enabled <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
29:26	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
25:24	Color Depth <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.									
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.									
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.									
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.									



XY_PAT_CHROMA_BLT_IMMEDIATE				
4..5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Destination Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
6 BR18	31:0	Transparency Color Low (Chroma-key Low = Pixel Greater or Equal)		
7 BR19	31:0	Transparency Color High (Chroma-key High = Pixel Less or Equal)		
8..n	31:0	Immediate Data		



XY_PIXEL_BLT

XY_PIXEL_BLT										
Source:	BlitterCS									
Length Bias:	2									
<p>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)).</p> <p>ROP field must specify pattern or fill with 0's or 1's. There is no source operand.</p> <p>Negative Stride (= Pitch) specified in the Setup command is Not Allowed</p>										
DWord	Bit	Description								
0 BR00	31:29	Client <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode				
	Default Value:	02h 2D Processor								
	Format:	Opcode								
	28:22	Instruction Target(Opcode) <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Default Value:</td> <td>24h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	24h	Format:	Opcode				
	Default Value:	24h								
	Format:	Opcode								
	21:12	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ									
11	Tiling Enable <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 45%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
Value	Name	Description								
0b	Tiling Disabled (Linear Blit)									
1b	Tiling Enabled	: Tile-X or Tile-Y.								
10:8	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
7:0	DWord Length <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Default Value:</td> <td>00h</td> </tr> </table>	Default Value:	00h							
Default Value:	00h									
1 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.								
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.								



XY_SCANLINES_BLT

XY_SCANLINES_BLT										
Source:	BlitterCS									
Length Bias:	2									
<p>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.</p> <p>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.</p> <p>Solid pattern should use the XY_SETUP_MONO_PATTERN_SL_BLT instruction.</p> <p>ROP field must specify pattern or fill with 0's or 1's. There is no source operand.</p>										
DWord	Bit	Description								
0 BR00	31:29	Client								
		Default Value: 02h 2D Processor Format: Opcode								
	28:22	Instruction Target(Opcode)								
		Default Value: 25h Format: Opcode								
	21:15	Reserved Format: MBZ								
	14:12	Pattern Horizontal Seed Pixel of the scan line to start on corresponding to DST X=0.								
	11	Tiling Enable								
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>		Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
Value		Name	Description							
0b	Tiling Disabled (Linear Blit)									
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 Length Default Value: 01h									
1 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.								
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.								
2 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.								
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.								



XY_SETUP_BLT

XY_SETUP_BLT								
Source:	BlitterCS							
Length Bias:	2							
<p>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.</p> <p>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).</p>								
DWord	Bit	Description						
0 BR00	31:29	Client <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode		
	Default Value:	02h 2D Processor						
	Format:	Opcode						
	28:22	Instruction Target(Opcod) <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>01h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	01h	Format:	Opcode		
	Default Value:	01h						
	Format:	Opcode						
	21:20	32 bpp Byte Mask <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td style="text-align: center;">x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	1xb	Write Alpha Channel	x1b	Write RGB Channel
	Value	Name						
	1xb	Write Alpha Channel						
	x1b	Write RGB Channel						
19:12	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ							
11	Tiling Enable <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Tiling Disabled (Linear Blit)</td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Tiling Enabled (Tile-X or Tile-Y)</td> </tr> </tbody> </table>	Value	Name	0b	Tiling Disabled (Linear Blit)	1b	Tiling Enabled (Tile-X or Tile-Y)	
Value	Name							
0b	Tiling Disabled (Linear Blit)							
1b	Tiling Enabled (Tile-X or Tile-Y)							
10:8	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ							
7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>08h</td> </tr> </table>	Default Value:	08h					
Default Value:	08h							
1 BR01	31	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ						



XY_SETUP_BLT											
	30	Clipping Enabled <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Disabled</td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
	29	Mono Source Transparency Mode <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Use Background</td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Transparency Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Use Background	1b	Transparency Enabled			
	Value	Name									
	0b	Use Background									
	1b	Transparency Enabled									
	28:26	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
25:24	Color Depth <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>8 Bit Color</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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	BR24	31:16 ClipRect Y1 Coordinate (Top) (30:16 = 15 bit positive number)									
		15:0 ClipRect X1 Coordinate (Left) (14:00 = 15 bit positive number)									
3	BR25	31:16 ClipRect Y2 Coordinate (Bottom) (30:16 = 15 bit positive number)									
		15:0 ClipRect X2 Coordinate (Right) (14:00 = 15 bit positive number)									
4.5	Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0 Destination Base Address <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0							
Format:	GraphicsAddress63-0										
6	BR05	31:0 Setup Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0] All									



XY_SETUP_BLT		
7 BR06	31:0	Setup Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0] (SLB and TB only)
8..9 28:06 are implemented. Note no NPO2 change here. The pattern data must be located in linear memory. The Setup Pattern Base Address for Color Pattern must always be Cache Line (64byte) aligned.	63:0	Setup Pattern Base Address for Color Pattern Format: GraphicsAddress63-0



XY_SETUP_CLIP_BLT

XY_SETUP_CLIP_BLT		
Source:	BlitterCS	
Length Bias:	2	
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 BR00	31:29	Client
		Default Value: 02h 2D Processor
		Format: Opcode
	28:22	Instruction Target(Opcode)
		Default Value: 03h
		Format: Opcode
	21:12	Reserved
Format: MBZ		
11	Tiling Enable	
	Value	Name
	0b	Tiling Disabled (Linear Blit)
	1b	Tiling Enabled (Tile-X or Tile-Y)
10:8	Reserved	
	Format: MBZ	
7:0	DWord Length	
	Default Value: 01h	
1 BR24	31:16	ClipRect Y1 Coordinate (Top) (30:16 = 15 bit positive number)
	15:0	ClipRect X1 Coordinate (Left) (14:00 = 15 bit positive number)
2 BR25	31:16	ClipRect Y2 Coordinate (Bottom) (30:16 = 15 bit positive number)
	15:0	ClipRect X2 Coordinate (Right) (14:00 = 15 bit positive number)



XY_SETUP_MONO_PATTERN_SL_BLT

XY_SETUP_MONO_PATTERN_SL_BLT								
Source:	BlitterCS							
Length Bias:	2							
<p>This setup instruction supplies common setup information including clipping coordinates used exclusively with the following instruction: XY_SCANLINE_BLT (SLB) - 1 scan line of monochrome pattern and destination are the only operands allowed.</p>								
DWord	Bit	Description						
0 BR00	31:29	Client						
		Default Value: 02h 2D Processor						
		Format: Opcode						
	28:22	Instruction Target(Opcode)						
		Default Value: 11h						
		Format: Opcode						
	21:20	32 bpp Byte Mask						
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td>x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>		Value	Name	1xb	Write Alpha Channel	x1b	Write RGB Channel	
Value		Name						
1xb	Write Alpha Channel							
x1b	Write RGB Channel							
19:12	Reserved							
	Format: MBZ							
11	Tiling Enable							
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> </tr> <tr> <td>1b</td> <td>Tiling Enabled (Tile-X or Tile-Y)</td> </tr> </tbody> </table>	Value	Name	0b	Tiling Disabled (Linear Blit)	1b	Tiling Enabled (Tile-X or Tile-Y)	
	Value	Name						
0b	Tiling Disabled (Linear Blit)							
1b	Tiling Enabled (Tile-X or Tile-Y)							
10:8	Reserved							
	Format: MBZ							
7:0	DWord Length							
	Default Value: 08h							
1 BR01	31	Solid Pattern Select (SLB and Pixel only)						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No Solid Pattern</td> </tr> <tr> <td>1</td> <td>Solid Pattern</td> </tr> </tbody> </table>	Value	Name	0	No Solid Pattern	1	Solid Pattern
		Value	Name					
0	No Solid Pattern							
1	Solid Pattern							



XY_SETUP_MONO_PATTERN_SL_BLT

	30	Clipping Enabled									
		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
	29	Reserved									
		Format: _____ MBZ									
	28	Mono Pattern Transparency Mode									
		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Use Background</td> </tr> <tr> <td>1b</td> <td>Transparency Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Use Background	1b	Transparency Enabled			
	Value	Name									
0b	Use Background										
1b	Transparency Enabled										
27:26	Reserved										
	Format: _____ MBZ										
25:24	Color Depth										
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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 BR24	31:16	ClipRect Y1 Coordinate (Top) (30:16 = 15 bit positive number)									
	15:0	ClipRect X1 Coordinate (Left) (14:00 = 15 bit positive number)									
3 BR25	31:16	ClipRect Y2 Coordinate (Bottom) (30:16 = 15 bit positive number)									
	15:0	ClipRect X2 Coordinate (Right) (14:00 = 15 bit positive number)									



XY_SETUP_MONO_PATTERN_SL_BLT				
4..5 BR05	63:0	Setup Destination Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
6 BR05	31:0	Setup Background Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0] All		
7 BR06	31:0	Setup Foreground Color 8 bit = [7:0], 16 bit = [15:0], 32 bit = [31:0] (SLB and TB only)		
8 BR20	31:0	DW0 (least significant) for a Monochrome Pattern		
9 BR21	31:0	DW1 (most significant) for a Monochrome Pattern		



XY_SRC_COPY_BLT

XY_SRC_COPY_BLT										
Source:	BlitterCS									
Length Bias:	2									
<p>This BLT instruction performs a color source copy where the only operands involved is a color source and destination of the same bit width.</p> <p>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.</p>										
DWord	Bit	Description								
0 BR00	31:29	Client <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode				
	Default Value:	02h 2D Processor								
	Format:	Opcode								
	28:22	Instruction Target(Opcode) <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Default Value:</td> <td>53h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	53h	Format:	Opcode				
	Default Value:	53h								
	Format:	Opcode								
	21:20	32bpp Byte Mask This field is only used for 32bpp. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td style="text-align: center;">[Default]</td> </tr> <tr> <td style="text-align: center;">1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td style="text-align: center;">x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	00b	[Default]	1xb	Write Alpha Channel	x1b	Write RGB Channel
	Value	Name								
	00b	[Default]								
	1xb	Write Alpha Channel								
x1b	Write RGB Channel									
19:16	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
15	Src Tiling Enable <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Tiling Disabled (Linear)</td> <td></td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear)		1b	Tiling Enabled	: Tile-X or Tile-Y.
Value	Name	Description								
0b	Tiling Disabled (Linear)									
1b	Tiling Enabled	: Tile-X or Tile-Y.								
14:12	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									



XY_SRC_COPY_BLT											
	11	Dest Tiling Enable <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
	Value	Name	Description								
	0b	Tiling Disabled (Linear Blit)									
	1b	Tiling Enabled	: Tile-X or Tile-Y.								
10:8	Reserved Format: _____ MBZ										
7:0	DWord Length Format: _____ =n <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>08h</td> <td></td> </tr> </tbody> </table>	Value	Name	08h							
Value	Name										
08h											
1 BR13	31	Reserved Format: _____ MBZ									
	30	Clipping Enabled <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
	29:26	Reserved Format: _____ MBZ									
25:24	Color Depth <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.									
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.									
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.									



XY_SRC_COPY_BLT		
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.
4..5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Destination Base Address Format: GraphicsAddress63-0
6 BR26	31:16	Source Y1 Coordinate (Top) 16 bit signed number.
	15:0	Source X1 Coordinate (Left) 16 bit signed number.
7 BR11	31:16	Reserved Format: MBZ
	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-Y and can be upto 128Kbytes (or 32KDwords).
8..9 Base address of the source surface: X=0, Y=0. When Src Tiling is enabled (Bit_15 enabled), this address must be 4KB-aligned. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Source Base Address Format: GraphicsAddress63-0



XY_SRC_COPY_CHROMA_BLT

XY_SRC_COPY_CHROMA_BLT										
Source:	BlitterCS									
Length Bias:	2									
<p>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.</p> <p>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.</p>										
DWord	Bit	Description								
0 BR00	31:29	Client <table border="1"> <tr> <td>Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode				
	Default Value:	02h 2D Processor								
	Format:	Opcode								
	28:22	Instruction Target(Opcode) <table border="1"> <tr> <td>Default Value:</td> <td>73h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	73h	Format:	Opcode				
	Default Value:	73h								
	Format:	Opcode								
	21:20	32bpp Byte Mask This field is only used for 32bpp. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>[Default]</td> </tr> <tr> <td>1xb</td> <td>Write Alpha Channel</td> </tr> <tr> <td>x1b</td> <td>Write RGB Channel</td> </tr> </tbody> </table>	Value	Name	00b	[Default]	1xb	Write Alpha Channel	x1b	Write RGB Channel
	Value	Name								
	00b	[Default]								
	1xb	Write Alpha Channel								
x1b	Write RGB Channel									
19:17	Transparency Range Mode (chroma-key)									
16	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
15	Src Tiling Enable <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear)		1b	Tiling Enabled	: Tile-X or Tile-Y.
Value	Name	Description								
0b	Tiling Disabled (Linear)									
1b	Tiling Enabled	: Tile-X or Tile-Y.								
14:12	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									



XY_SRC_COPY_CHROMA_BLT											
	11	Dest Tiling Enable <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td>1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
	Value	Name	Description								
	0b	Tiling Disabled (Linear Blit)									
	1b	Tiling Enabled	: Tile-X or Tile-Y.								
10:8	Reserved Format: _____ MBZ										
7:0	DWord Length <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0Ah</td> <td></td> </tr> </tbody> </table>	Value	Name	0Ah							
Value	Name										
0Ah											
1 BR13	31	Reserved Format: _____ MBZ									
	30	Clipping Enabled <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	0b	Disabled	1b	Enabled			
	Value	Name									
	0b	Disabled									
	1b	Enabled									
	29:26	Reserved Format: _____ MBZ									
25:24	Color Depth <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color</td> </tr> <tr> <td>01b</td> <td>16 Bit Color(565)</td> </tr> <tr> <td>10b</td> <td>16 Bit Color(1555)</td> </tr> <tr> <td>11b</td> <td>32 Bit Color</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color	01b	16 Bit Color(565)	10b	16 Bit Color(1555)	11b	32 Bit Color
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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.									
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.									
3 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.									
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.									



XY_SRC_COPY_CHROMA_BLT				
4..5 Base address of the destination surface: X=0, Y=0. When Tiling is enabled (Bit_11 enabled), this address is limited to 4Kbytes. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Destination Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
6 BR26	31:16	Source Y1 Coordinate (Top) 16 bit signed number.		
	15:0	Source X1 Coordinate (Left) 16 bit signed number.		
7 BR11	31:16	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
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-Y and can be upto 128Kbytes (or 32KWords).			
8..9 Base address of the source surface: X=0, Y=0. When Src Tiling is enabled (Bit_15 enabled), this address must be 4KB-aligned. When Tiling is not enabled, this address is to be CL (64byte) aligned.	63:0	Source Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>GraphicsAddress63-0</td> </tr> </table>	Format:	GraphicsAddress63-0
Format:	GraphicsAddress63-0			
10 BR18	31:0	Transparency Color Low (Chroma-key Low = Pixel Greater or Equal)		
11 BR19	31:0	Transparency Color High (Chroma-key High = Pixel Less or Equal)		



XY_TEXT_BLT

XY_TEXT_BLT										
Source:	BlitterCS									
Length Bias:	2									
<p>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.</p> <p>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.</p> <p>Source expansion color registers are always in the SETUP_BLT.</p> <p>Negative Stride (= Pitch) is NOT ALLOWED.</p>										
DWord	Bit	Description								
0 BR00	31:29	Client <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Default Value:</td> <td>02h 2D Processor</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	02h 2D Processor	Format:	Opcode				
	Default Value:	02h 2D Processor								
	Format:	Opcode								
	28:22	Instruction Target(Opcode) <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Default Value:</td> <td>26h</td> </tr> <tr> <td>Format:</td> <td>Opcode</td> </tr> </table>	Default Value:	26h	Format:	Opcode				
	Default Value:	26h								
	Format:	Opcode								
	21:17	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
	Format:	MBZ								
	16	Bit / Byte Packed Byte packed is for the NT driver. <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Bit</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Byte</td> </tr> </tbody> </table>	Value	Name	0	Bit	1	Byte		
	Value	Name								
0	Bit									
1	Byte									
15:12	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
11	Tiling Enable <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 45%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td>Tiling Disabled (Linear Blit)</td> <td></td> </tr> <tr> <td style="text-align: center;">1b</td> <td>Tiling Enabled</td> <td>: Tile-X or Tile-Y.</td> </tr> </tbody> </table>	Value	Name	Description	0b	Tiling Disabled (Linear Blit)		1b	Tiling Enabled	: Tile-X or Tile-Y.
Value	Name	Description								
0b	Tiling Disabled (Linear Blit)									
1b	Tiling Enabled	: Tile-X or Tile-Y.								
10:8	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
7:0	DWord Length <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Default Value:</td> <td>03h</td> </tr> </table>	Default Value:	03h							
Default Value:	03h									



XY_TEXT_BLT		
1 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.
2 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.
3..4 Address of the first byte on a scan line corresponding to source X1, Y1. Note no NPO2 change here. The source address must always be Cache Line (64byte) aligned.	63:0	Source Address Format: GraphicsAddress63-0



XY_TEXT_IMMEDIATE_BLT

XY_TEXT_IMMEDIATE_BLT			
Source:	BlitterCS		
Length Bias:	2		
<p>This instruction allows the Driver to send data through the instruction stream that eliminates the read latency of reading a source from memory.</p> <p>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.</p> <p>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.</p> <p>Source expansion color registers are always in the SETUP_BLT. NEGATIVE STRIDE (= PITCH) IS NOT ALLOWED.</p>			
DWord	Bit	Description	
0 BR00	31:29	Client	
		Default Value:	02h 2D Processor
		Format:	Opcode
	28:22	Instruction Target(Opcode)	
		Default Value:	31h
		Format:	Opcode
	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 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 BR22	31:16	Destination Y1 Coordinate (Top) 16 bit signed number.
	15:0	Destination X1 Coordinate (Left) 16 bit signed number.
2 BR23	31:16	Destination Y2 Coordinate (Bottom) 16 bit signed number.
	15:0	Destination X2 Coordinate (Right) 16 bit signed number.
3..n	31:0	Immediate Data