



Intel® Open Source HD Graphics, Intel Iris™ Graphics, and Intel Iris™ Pro Graphics

Programmer's Reference Manual

For the 2015 - 2016 Intel Core™ Processors, Celeron™ Processors, and Pentium™ Processors based on the "Skylake" Platform

Volume 2d: Command Reference: Structures

May 2016, 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 © 2016, Intel Corporation. All rights reserved.

Table of Contents

3DSTATE_CONSTANT_ALL_DATA	1
3DSTATE_CONSTANT(Body)	2
A32 Buffer Base Address Message Header Control	5
A32 Scaled Header Present Message Descriptor Control Field	6
A32 Sideband Scale and Offset Enable Message Descriptor Control Field	7
A64 Data Size Message Descriptor Control Field	8
A64 Dual Oword Block Message Header	9
A64 Hword Block Message Header	10
A64 Hword Data Blocks Message Descriptor Control Field	11
A64 Oword Block Message Header	12
A64 Oword Data Blocks Message Descriptor Control Field	13
A64 Oword Dual Data Blocks Message Descriptor Control Field	14
A64 Scaled Data Port 2 Message Header	15
A64 Scaled Header Present Message Descriptor Control Field	16
A64 Sideband Scale and Offset Enable Message Descriptor Control Field	17
AddrSubRegNum	18
Any Binding Table Index Message Descriptor Control Field	19
Atomic Float Binary Operation Message Descriptor Control Field	20
Atomic Float Trinary Operation Message Descriptor Control Field	21
Atomic Integer Binary Operation Message Descriptor Control Field	22
Atomic Integer Trinary Operation Message Descriptor Control Field	23
Atomic Integer Unary Operation Message Descriptor Control Field	24
Audio Power State Format	25
AVC CABAC	26
AVC CAVLC	27
AVS_Inline_DMEN	29
BaseAddress4KByteAligned	32
BCS Hardware-Detected Error Bit Definitions	33
BINDING_TABLE_EDIT_ENTRY	34
BINDING_TABLE_STATE	35
Bit Definition for Interrupt Control Registers - Blitter	36
Bit Definition for Interrupt Control Registers - Media#1	38

Bit Definition for Interrupt Control Registers - Media#2.....40

Bit Definition for Interrupt Control Registers - Render.....42

Bit Definition for Interrupt Control Registers - Video Enhancement.....44

BLEND_STATE.....46

BLEND_STATE_ENTRY.....48

Block Dimensions Message Header Control.....52

Block Message Header.....53

BR00 - BLT Opcode and Control.....54

BR01 - Setup BLT Raster OP, Control, and Destination Offset.....58

BR05 - Setup Expansion Background Color.....61

BR06 - Setup Expansion Foreground Color.....62

BR07 - Setup Blit Color Pattern Address Lower Order Address bits.....63

BR09 - Destination Address Lower Order Address Bits.....64

BR11 - BLT Source Pitch (Offset).....65

BR12 - Source Address Lower order Address bits.....66

BR13 - BLT Raster OP, Control, and Destination Pitch.....67

BR14 - Destination Width and Height.....70

BR15 - Color Pattern Address Lower order Address bits.....71

BR16 - Pattern Expansion Background and Solid Pattern Color.....72

BR17 - Pattern Expansion Foreground Color.....73

BR18 - Source Expansion Background and Destination Color.....74

BR19 - Source Expansion Foreground Color.....75

BR27 - Destination Address Higher Order Address.....76

BR28 - Source Address Higher order Address.....77

BR29 - Color Pattern Address Higher order Address.....78

BR30 - Setup Blit Color Pattern Address Higher Order Address.....79

Byte Masked Media Block Message Header.....80

Byte Masked Media Block Message Header Control.....81

CC_VIEWPORT.....83

Channel Mask Message Descriptor Control Field.....84

Channel Mode Message Descriptor Control Field.....85

Clock Gating Disable Format.....86

Clock Gating Disable Format.....87

COLOR_CALC_STATE.....88

COLOR_PROCESSING_STATE - ACE State	90
COLOR_PROCESSING_STATE - CSC State	95
COLOR_PROCESSING_STATE - PROCAMP State	98
COLOR_PROCESSING_STATE - STD/STE State	99
COLOR_PROCESSING_STATE - TCC State	110
Color Calculator State Pointer Message Header Control	114
Color Code Message Header Control.....	115
Context Descriptor Format.....	116
Context Status.....	120
CSC COEFFICIENT FORMAT	122
Data Port 0 Message Types	123
Data Port 1 Message Types	124
Data Port 2 Extended Message Descriptor	126
Data Port 2 Message Types	127
Data Port Bindless Surface Extended Message Descriptor	128
Data Size Message Descriptor Control Field.....	129
Depth Clear Value Format.....	130
Display Engine Render Response Message Definition	131
DstRegNum.....	133
DstSubRegNum.....	134
Dword Data Payload Register.....	135
Dword SIMD4x2 Atomic CMPWR Message Data Payload	136
Dword SIMD4x2 Atomic Operation Message Data Payload.....	137
Dword SIMD4x2 Data Payload	138
Dword SIMD8 Atomic Operation CMPWR Message Data Payload	139
Dword SIMD8 Data Payload.....	140
Dword SIMD16 Atomic Operation CMPWR Message Data Payload	141
Dword SIMD16 Data Payload	142
DX9_CONSTANTB_ENTRY.....	143
DX9_CONSTANTF_ENTRY	144
DX9_CONSTANTI_ENTRY	145
Encoder Control State Parameters0.....	146
Encoder Statistics Format	153
EU_INSTRUCTION_BASIC_ONE_SRC	157

EU_INSTRUCTION_BASIC_THREE_SRC 158

EU_INSTRUCTION_BASIC_TWO_SRC 161

EU_INSTRUCTION_BRANCH_CONDITIONAL 162

EU_INSTRUCTION_BRANCH_ONE_SRC 164

EU_INSTRUCTION_BRANCH_TWO_SRC 165

EU_INSTRUCTION_COMPACT_THREE_SRC 166

EU_INSTRUCTION_COMPACT_TWO_SRC 169

EU_INSTRUCTION_CONTROLS_A 174

EU_INSTRUCTION_CONTROLS_B 177

EU_INSTRUCTION_CONTROLS 179

EU_INSTRUCTION_HEADER 180

EU_INSTRUCTION_ILLEGAL 181

EU_INSTRUCTION_MATH 182

EU_INSTRUCTION_NOP 183

EU_INSTRUCTION_OPERAND_CONTROLS 184

EU_INSTRUCTION_OPERAND_DST_ALIGN1 186

EU_INSTRUCTION_OPERAND_DST_ALIGN16 187

EU_INSTRUCTION_OPERAND_SEND_MSG 188

EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1 189

EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16 190

EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC 191

EU_INSTRUCTION_SEND 192

EU_INSTRUCTION_SENDS 193

EU_INSTRUCTION_SOURCES_IMM32 195

EU_INSTRUCTION_SOURCES_REG 196

EU_INSTRUCTION_SOURCES_REG_IMM 197

EU_INSTRUCTION_SOURCES_REG_REG 198

Execution_Unit_Extended_Message_Descriptor 200

Extended Message Descriptor - Execution Unit 201

Extended Message Descriptor Render Target 202

Extended Message Descriptor - Sampling Engine 203

ExtMsgDescpt 204

ExtMsgDescptImmediate 206

FFTID Message Header Control 208

Filter_Coefficient	209
Filter_Coefficients	210
FrameDeltaQp	211
FrameDeltaQpRange	212
FunctionControl	213
GATHER_CONSTANT_ENTRY	214
GTC Interrupt Bit Definition	215
Half Precision Dual Source SIMD8 Message Data Payload Register	216
Half Precision OM Replicated SIMD16 Render Target Data Payload	218
Half Precision OM S0A SIMD8 Render Target Data Payload	219
Half Precision OM S0A SIMD16 Render Target Data Payload	220
Half Precision OM SIMD8 Dual Source Render Target Data Payload	221
Half Precision OM SIMD8 Render Target Data Payload	222
Half Precision OM SIMD16 Render Target Data Payload	223
Half Precision OS OM S0A SIMD8 Render Target Data Payload	224
Half Precision OS OM SIMD8 Dual Source Render Target Data Payload	226
Half Precision OS OM SIMD8 Render Target Data Payload	227
Half Precision OS S0A SIMD8 Render Target Data Payload	228
Half Precision OS SIMD8 Dual Source Render Target Data Payload	229
Half Precision OS SIMD8 Render Target Data Payload	230
Half Precision OS SZ OM S0A SIMD8 Render Target Data Payload	231
Half Precision OS SZ OM SIMD8 Dual Source Render Target Data Payload	233
Half Precision OS SZ OM SIMD8 Render Target Data Payload	235
Half Precision OS SZ S0A SIMD8 Render Target Data Payload	237
Half Precision OS SZ SIMD8 Dual Source Render Target Data Payload	239
Half Precision OS SZ SIMD8 Render Target Data Payload	240
Half Precision Replicated Pixel Render Target Data Payload Register	241
Half Precision Replicated SIMD16 Render Target Data Payload	242
Half Precision S0A SIMD8 Render Target Data Payload	243
Half Precision S0A SIMD16 Render Target Data Payload	244
Half Precision SIMD8 Dual Source Render Target Data Payload	245
Half Precision SIMD8 Message Data Payload Register	246
Half Precision SIMD8 Render Target Data Payload	248
Half Precision SIMD16 Message Data Payload Register	249

Half Precision SIMD16 Render Target Data Payload 251

Half Precision SZ OM S0A SIMD8 Render Target Data Payload 252

Half Precision SZ OM S0A SIMD16 Render Target Data Payload 254

Half Precision SZ OM SIMD8 Dual Source Render Target Data Payload 256

Half Precision SZ OM SIMD8 Render Target Data Payload 257

Half Precision SZ OM SIMD16 Render Target Data Payload 258

Half Precision SZ S0A SIMD8 Render Target Data Payload 260

Half Precision SZ S0A SIMD16 Render Target Data Payload 261

Half Precision SZ SIMD8 Dual Source Render Target Data Payload 263

Half Precision SZ SIMD8 Render Target Data Payload 264

Half Precision SZ SIMD16 Render Target Data Payload 265

Hardware-Detected Error Bit Definitions 266

Hardware Status Page Layout 267

HCP_PAK_INSERT_OBJECT_INDIRECT_PAYLOAD 272

Header Forbidden Message Descriptor Control Field 273

Header Present Message Descriptor Control Field 274

Header Required Message Descriptor Control Field 275

HW Generated BINDING_TABLE_STATE 276

Hword 1 Block Data Payload 277

Hword 2 Block Data Payload 278

Hword 4 Block Data Payload 279

Hword 8 Block Data Payload 280

Hword Channel Mode Message Header Control 282

Hword Register Blocks Message Descriptor Control Field 283

Ignored Message Header 284

Inline Data Description for MFD_AVC_BSD_Object 285

Inline Data Description - VP8 PAK OBJECT 293

INTERFACE_DESCRIPTOR_DATA 298

INTERRUPT 304

Invalidate After Read Message Descriptor Control Field 306

JPEG 307

LOD Message Address Payload Control 308

Lower Oword Block Data Payload 309

MEDIA_SURFACE_STATE 310

MEMORY_OBJECT_CONTROL_STATE	321
MemoryAddressAttributes	322
Merged Media Block Message Header	323
Merged Media Block Message Header Control	324
Message Descriptor - Render Target Write	326
Message Descriptor - Sampling Engine	328
MFD_MPEG2_BSD_OBJECT Inline Data Description	330
MPEG2	332
MPEG4-2_Inline_DMED	333
MSAA Sample Number Message Address Control	339
MsgDescpt31	340
Normal Media Block Message Header	341
Normal Media Block Message Header Control	342
oMask Message Data Payload Register	344
OM Replicated SIMD16 Render Target Data Payload	346
OM S0A SIMD8 Render Target Data Payload	347
OM S0A SIMD16 Render Target Data Payload	348
OM SIMD8 Dual Source Render Target Data Payload	350
OM SIMD8 Render Target Data Payload	352
OM SIMD16 Render Target Data Payload	353
OS OM S0A SIMD8 Render Target Data Payload	355
OS OM SIMD8 Dual Source Render Target Data Payload	357
OS OM SIMD8 Render Target Data Payload	359
OS S0A SIMD8 Render Target Data Payload	360
OS SIMD8 Dual Source Render Target Data Payload	361
OS SIMD8 Render Target Data Payload	363
OS SZ OM S0A SIMD8 Render Target Data Payload	364
OS SZ OM SIMD8 Dual Source Render Target Data Payload	366
OS SZ OM SIMD8 Render Target Data Payload	368
OS SZ S0A SIMD8 Render Target Data Payload	369
OS SZ SIMD8 Dual Source Render Target Data Payload	370
OS SZ SIMD8 Render Target Data Payload	372
Oword 1 Dual Block Data Payload	373
Oword 2 Block Data Payload	374

Oword 4 Block Data Payload 375

Oword 4 Dual Block Data Payload 376

Oword 8 Block Data Payload 377

Oword A64 SIMD4x2 Atomic CMPWR16B Message Data Payload 378

Oword A64 SIMD4x2 Atomic Operation Return Data Message Data Payload 379

Oword A64 SIMD8 Atomic Operation CMPWR16B Message Data Payload..... 380

Oword Data Blocks Message Descriptor Control Field 382

Oword Data Payload Register..... 383

Oword Dual Data Blocks Message Descriptor Control Field..... 384

PALETTE_ENTRY 385

Performance Counter Report Format 101b 386

Per Thread Scratch Space Message Header Control 389

Pixel Masked Media Block Message Header 390

Pixel Masked Media Block Message Header Control 391

Pixel Sample Mask Message Header Control 393

Pixel Sample Mask Render Target Message Header Control..... 394

Power Clock State Format 395

PPHWSP_LAYOUT 397

Qword A64 SIMD4x2 Atomic CMPWR Message Data Payload 399

Qword Data Payload Register..... 400

Qword SIMD4x2 Atomic CMPWR8B Message Data Payload..... 401

Qword SIMD4x2 Atomic Operation Message Data Payload..... 402

Qword SIMD8 Atomic Operation CMPWR8B Message Data Payload 403

Qword SIMD8 Atomic Operation CMPWR Message Data Payload 404

Qword SIMD8 Atomic Operation Return Data Message Data Payload..... 405

Qword SIMD8 Data Payload 406

Qword SIMD16 Atomic Operation CMPWR8B Message Data Payload..... 407

Qword SIMD16 Atomic Operation Return Data Message Data Payload 409

Qword SIMD16 Data Payload 410

Read-Only Data Port Message Types 411

Read Surface Info 32-Bit Address Payload 412

Read Surface Info Data Payload 413

RENDER_SURFACE_STATE..... 415

Render Data Port Message Types 445

Render Target Index Message Header Control	446
Render Target Message Header	447
Render Target Message Header Control	449
Replicated Pixel Render Target Data Payload Register	451
Replicated SIMD16 Render Target Data Payload	452
Reversed SIMD Mode 2 Message Descriptor Control Field	453
RoundingPrecisionTable_3_Bits	454
S0A SIMD8 Render Target Data Payload	455
S0A SIMD16 Render Target Data Payload	456
SAMPLER_BORDER_COLOR_STATE	458
SAMPLER_INDIRECT_STATE_BORDER_COLOR	460
SAMPLER_INDIRECT_STATE	463
SAMPLER_STATE_8x8_AVS_COEFFICIENTS	466
SAMPLER_STATE_8x8_AVS	470
SAMPLER_STATE	484
SCISSOR_RECT	495
Scratch Hword Block Message Header	496
SF_CLIP_VIEWPORT	497
SF_OUTPUT_ATTRIBUTE_DETAIL	499
SFC_8x8_AVS_COEFFICIENTS	501
SIMD4x2 MSAA Typed Surface 32-Bit Address Payload	504
SIMD4x2 Typed Surface 32-Bit Address Payload	506
SIMD4x2 Untyped BUFFER Surface 32-Bit Address Payload	507
SIMD4x2 Untyped BUFFER Surface 64-Bit Address Payload	508
SIMD4x2 Untyped STRBUF Surface 32-Bit Address Payload	509
SIMD4x2 32-Bit Address Payload	510
SIMD8 Dual Source Render Target Data Payload	511
SIMD8 LOD Message Address Payload Control	513
SIMD8 MSAA Typed Surface 32-Bit Address Payload	514
SIMD8 Render Target Data Payload	516
SIMD8 Typed Surface 32-Bit Address Payload	517
SIMD8 Untyped BUFFER Surface 32-Bit Address Payload	518
SIMD8 Untyped BUFFER Surface 64-Bit Address Payload	519
SIMD8 Untyped STRBUF Surface 32-Bit Address Payload	520

SIMD16 Render Target Data Payload 521

SIMD16 Untyped BUFFER Surface 32-Bit Address Payload 523

SIMD16 Untyped BUFFER Surface 64-Bit Address Payload 524

SIMD16 Untyped STRBUF Surface 32-Bit Address Payload 525

SIMD 32-Bit Address Payload Control 526

SIMD 64-Bit Address Payload Control 527

SIMD8 32-Bit Address Payload 528

SIMD8 64-Bit Address Payload 529

SIMD16 32-Bit Address Payload 530

SIMD16 64-Bit Address Payload 531

SIMD Mode 2 Message Descriptor Control Field 532

SIMD Mode 3 Message Descriptor Control Field 533

SLM Surface Pixel Mask Message Header 534

Slot Group 2 Message Descriptor Control Field 535

Slot Group 3 Message Descriptor Control Field 536

Slot Group Select Render Cache Message Descriptor Control Field 537

SO_DECL 538

SO_DECL_ENTRY 540

Split_coding_unit_flags 541

SplitBaseAddress4KByteAligned 542

SplitBaseAddress64ByteAligned 543

SrcRegNum 544

SrcSubRegNum 545

SRD Interrupt Bit Definition 546

Stateless Binding Table Index Message Descriptor Control Field 547

Stateless Block Message Header 548

Stateless Scaled Data Port 2 Message Header 549

Stateless Scaled Data Port 2 Pixel Mask Message Header 550

Stateless Surface Message Header 551

Stateless Surface Pixel Mask Message Header 552

Stencil Message Data Payload Register 553

Subset Atomic Integer Ternary Operation Message Descriptor Control Field 555

Subset Reversed SIMD Mode 2 Message Descriptor Control Field 556

Subset SIMD Mode 2 Message Descriptor Control Field 557

Subset SIMD Mode 3 Message Descriptor Control Field	558
Subspan Render Target Message Header Control	559
Surface Binding Table Index Message Descriptor Control Field	560
Surface or Stateless Binding Table Index Message Descriptor Control Field	561
Surface Pitch Message Header Control	562
Surface Pixel Mask Message Header	563
SW Generated BINDING_TABLE_STATE	564
SZ OM S0A SIMD8 Render Target Data Payload	565
SZ OM S0A SIMD16 Render Target Data Payload	567
SZ OM SIMD8 Dual Source Render Target Data Payload	569
SZ OM SIMD8 Render Target Data Payload	571
SZ OM SIMD16 Render Target Data Payload	572
SZ S0A SIMD8 Render Target Data Payload	574
SZ S0A SIMD16 Render Target Data Payload	575
SZ SIMD8 Dual Source Render Target Data Payload	577
SZ SIMD8 Render Target Data Payload	579
SZ SIMD16 Render Target Data Payload	580
Thread EOT Message Descriptor	582
Thread Spawn Message Descriptor	583
TileW SIMD8 Data Control Dword	585
TileW SIMD8 Data Payload	586
Transpose Message Header	587
Untyped Write Channel Mask Message Descriptor Control Field	588
Upper Oword Block Data Payload	589
VC1	590
VCS Hardware-Detected Error Bit Definitions	591
VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS	592
VEBOX_ACE_LACE_STATE	593
VEBOX_ALPHA_AOI_STATE	600
VEBOX_CAPTURE_PIPE_STATE	602
VEBOX_CCM_STATE	606
VEBOX_Ch_Dir_Filter_Coefficient	609
VEBOX_CSC_STATE	610
VEBOX_DNDI_STATE	614



VEBOX_Filter_Coefficient	623
VEBOX_FORWARD_GAMMA_CORRECTION_STATE	624
VEBOX_FRONT_END_CSC_STATE	625
VEBOX_GAMUT_STATE	628
VEBOX_IECP_STATE	645
VEBOX_PROCAMP_STATE	647
VEBOX_RGB_TO_GAMMA_CORRECTION	648
VEBOX_STD_STE_STATE	649
VEBOX_TCC_STATE	665
VEBOX_VERTEX_TABLE	670
VECS Hardware-Detected Error Bit Definitions	673
VERTEX_BUFFER_STATE	674
VERTEX_ELEMENT_STATE	676
Vertical Line Stride Override Message Descriptor Control Field	680
VFE_STATE_EX	681
VP8 Encoder StreamOut Format	683

3DSTATE_CONSTANT_ALL_DATA

3DSTATE_CONSTANT_ALL_DATA				
Source:	RenderCS			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0..1	63:5	<p>Pointer To Constant Buffer</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[63:5]ConstantBuffer</td> </tr> </table> <p>The value of this field is the virtual address of the location of the push constant buffer. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p> <p style="text-align: center;">Programming Notes</p> <p>Constant buffers must be allocated in linear (not tiled) graphics memory.</p>	Format:	GraphicsAddress[63:5]ConstantBuffer
	Format:	GraphicsAddress[63:5]ConstantBuffer		
4:0	<p>Constant Buffer Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>This field specifies the length of the constant data to be loaded from memory in 256-bit units.</p> <p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> The sum of the read length fields for all pointers must be less than or equal to the size of 64 Zero means there no data to fetch for this buffer pointer. 	Format:	U5	
Format:	U5			

3DSTATE_CONSTANT(Body)

3DSTATE_CONSTANT(Body)				
Source:	RenderCS			
Size (in bits):	320			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	<p>Constant Buffer 1 Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U16 read length</td> </tr> </table> <p>This field specifies the length of the constant data to be loaded from memory in 256-bit units.</p> <p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> The sum of all four read length fields must be less than or equal to the size of 64 Setting the value of the register to zero will disable buffer 1. If disabled, the Pointer to Constant Buffer 1 must be programmed to zero. <p>if gather constant are enabled, this field must be non-zero if a there was a preceding corresponding 3DSTATE_GATHER_CONSTANT_*, otherwise this field must be zero.</p>	Format:	U16 read length
	Format:	U16 read length		
15:0	<p>Constant Buffer 0 Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U16 read length</td> </tr> </table> <p>This field specifies the length of the constant data to be loaded from memory in 256-bit units.</p> <p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> The sum of all four read length fields must be less than or equal to the size of 64 Setting the value of the register to zero will disable buffer 0. If disabled, the Pointer to Constant Buffer 0 must be programmed to zero. 	Format:	U16 read length	
Format:	U16 read length			
1	31:16	<p>Constant Buffer 3 Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U16 read length</td> </tr> </table> <p>This field specifies the length of the constant data to be loaded from memory in 256-bit units.</p> <p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> The sum of all four read length fields must be less than or equal to the size of 64 Setting the value of the register to zero will disable buffer 3. If disabled, the Pointer to Constant Buffer 3 must be programmed to zero. 	Format:	U16 read length
	Format:	U16 read length		

3DSTATE_CONSTANT(Body)								
	15:0	<p>Constant Buffer 2 Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U16 read length</td> </tr> </table> <p>This field specifies the length of the constant data to be loaded from memory in 256-bit units.</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2"> <ul style="list-style-type: none"> The sum of all four read length fields must be less than or equal to the size of 64 Setting the value of the register to zero will disable buffer 2. If disabled, the Pointer to Constant Buffer 2 must be programmed to zero. </td> </tr> </table>	Format:	U16 read length	Programming Notes		<ul style="list-style-type: none"> The sum of all four read length fields must be less than or equal to the size of 64 Setting the value of the register to zero will disable buffer 2. If disabled, the Pointer to Constant Buffer 2 must be programmed to zero. 	
	Format:	U16 read length						
Programming Notes								
<ul style="list-style-type: none"> The sum of all four read length fields must be less than or equal to the size of 64 Setting the value of the register to zero will disable buffer 2. If disabled, the Pointer to Constant Buffer 2 must be programmed to zero. 								
2..3	63:5	Reserved						
	4:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ							
4..5	63:5	<p>Pointer To Constant Buffer 1</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[63:5]ConstantBuffer</td> </tr> </table> <p>This field points to the location of Constant Buffer 1.</p> <p>If gather constants are enabled This field is an offset of constant Buffer1 from the Gather Pool BASE ADDRESS.</p> <p>If gather constants is disabled, the value of this field is the virtual address of the location of the push constant buffer. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">Constant buffers must be allocated in linear (not tiled) graphics memory.</td> </tr> </table>	Format:	GraphicsAddress[63:5]ConstantBuffer	Programming Notes		Constant buffers must be allocated in linear (not tiled) graphics memory.	
		Format:	GraphicsAddress[63:5]ConstantBuffer					
	Programming Notes							
Constant buffers must be allocated in linear (not tiled) graphics memory.								
4:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ							
6..7	63:5	<p>Pointer To Constant Buffer 2</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[63:5]ConstantBuffer</td> </tr> </table> <p>The value of this field is the virtual address of the location of the push constant buffer 2. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">Constant buffers must be allocated in linear (not tiled) graphics memory.</td> </tr> </table>	Format:	GraphicsAddress[63:5]ConstantBuffer	Programming Notes		Constant buffers must be allocated in linear (not tiled) graphics memory.	
		Format:	GraphicsAddress[63:5]ConstantBuffer					
	Programming Notes							
Constant buffers must be allocated in linear (not tiled) graphics memory.								
4:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ							

3DSTATE_CONSTANT(Body)								
8..9	63:5	<p>Pointer To Constant Buffer 3</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>GraphicsAddress[63:5]ConstantBuffer</td> </tr> </table> <p>The value of this field is the virtual address of the location of the push constant buffer 3. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">Constant buffers must be allocated in linear (not tiled) graphics memory.</td> </tr> </table>	Format:	GraphicsAddress[63:5]ConstantBuffer	Programming Notes		Constant buffers must be allocated in linear (not tiled) graphics memory.	
	Format:	GraphicsAddress[63:5]ConstantBuffer						
Programming Notes								
Constant buffers must be allocated in linear (not tiled) graphics memory.								
	4:0	<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							

A32 Buffer Base Address Message Header Control

MHC_A32_BBA - A32 Buffer Base Address Message Header Control				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:0	<p>Buffer Base Address Offset</p> <table border="1"> <tr> <td>Format:</td> <td>GeneralStateOffset[31:0]</td> </tr> </table> <p>Specifies the base address offset page [31:10] for A32 stateless messages.</p>	Format:	GeneralStateOffset[31:0]
Format:	GeneralStateOffset[31:0]			

A32 Scaled Header Present Message Descriptor Control Field

MDC_A32_MHP - A32 Scaled Header Present Message Descriptor Control Field													
Source:	BSpec												
Size (in bits):	1												
Default Value:	0x00000000												
DWord	Bit	Description											
0	0	<p>Message Header Present</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> </table> <p>Specifies if the message uses the optional message header to modify the access type and address calculation, in combination with the MDC_A32_SSO field.</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;">0h</td> <td style="text-align: center;">No</td> <td>Message header is not present. The Sideband Scale Offset field from the Message Descriptor are used as offsets with the Address Payload.</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">Yes</td> <td>Message header is present. It specifies a Base Address Offset and a Scale pitch to use with each of the offsets from the Address Payload.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>The access is an SLM access if the Sideband Scale Offset is enabled when the Message Header is not present; otherwise the access is a A32 access.</p>	Format:	Enumeration	Value	Name	Description	0h	No	Message header is not present. The Sideband Scale Offset field from the Message Descriptor are used as offsets with the Address Payload.	1h	Yes	Message header is present. It specifies a Base Address Offset and a Scale pitch to use with each of the offsets from the Address Payload.
Format:	Enumeration												
Value	Name	Description											
0h	No	Message header is not present. The Sideband Scale Offset field from the Message Descriptor are used as offsets with the Address Payload.											
1h	Yes	Message header is present. It specifies a Base Address Offset and a Scale pitch to use with each of the offsets from the Address Payload.											

A32 Sideband Scale and Offset Enable Message Descriptor Control Field

MDC_A32_SBSO - A32 Sideband Scale and Offset Enable Message Descriptor Control Field					
Source:	BSpec				
Size (in bits):	8				
Default Value:	0x00000000				
DWord	Bit	Description			
0	7	<p>Sideband Offset Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>When set, enables adding the 16-bit offset from the Sideband to all the offsets in the Address Payload for the A32 or SLM access. The 16-bit Sideband Offset is specified in the extended function control field in the SEND instruction.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </table> <p>If set and a Message Header is not present, then the access is an SLM access and the Sideband Offset and Scale are used as offsets with the Address Payload. Otherwise, the access is an A32 access, and either or both of the Message Header and Sideband Offsets are added to the offsets in the Address Payload.</p>	Format:	Enable	Programming Notes
	Format:	Enable			
Programming Notes					
6:0	<p>Scale</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U7</td> </tr> </table> <p>Specifies the scale pitch to be used for SLM messages as (#bytes-1).</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </table> <p>If this is a A32 message, this field is ignored and the scale pitch is supplied by the Message Header. If this is an A32 message and no Message Header present, then the scale is 0 (i.e. pitch = 1 byte).</p>	Format:	U7	Programming Notes	
Format:	U7				
Programming Notes					

A64 Data Size Message Descriptor Control Field

MDC_A64_DS - A64 Data Size Message Descriptor Control Field																							
Source:	BSpec																						
Size (in bits):	2																						
Default Value:	0x00000000																						
DWord	Bit	Description																					
0	1:0	Data Size Format: Enumeration Specifies the number of data elements to be read or written																					
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>DE1</td> <td>1 data element (B, DW, QW)</td> <td></td> </tr> <tr> <td>01h</td> <td>DE2</td> <td>2 data elements (B, DW, QW)</td> <td></td> </tr> <tr> <td>02h</td> <td>DE4</td> <td>4 data elements (B, DW, QW)</td> <td></td> </tr> <tr> <td>03h</td> <td>DE8</td> <td>8 data elements (B, DW, QW)</td> <td>This setting is supported for DW and QW but not for B. For bytes, the maximum number of data elements is 4.</td> </tr> </tbody> </table>	Value	Name	Description	Programming Notes	00h	DE1	1 data element (B, DW, QW)		01h	DE2	2 data elements (B, DW, QW)		02h	DE4	4 data elements (B, DW, QW)		03h	DE8	8 data elements (B, DW, QW)	This setting is supported for DW and QW but not for B. For bytes, the maximum number of data elements is 4.	
Value	Name	Description	Programming Notes																				
00h	DE1	1 data element (B, DW, QW)																					
01h	DE2	2 data elements (B, DW, QW)																					
02h	DE4	4 data elements (B, DW, QW)																					
03h	DE8	8 data elements (B, DW, QW)	This setting is supported for DW and QW but not for B. For bytes, the maximum number of data elements is 4.																				
		Restriction The number of elements is constrained by SIMD Mode and Data Width. The max data payload limit is 256B: 2 elements SIMD16 QW, 4 elements SIMD16 DW, or 4 elements SIMD8 QW.																					

A64 Dual Oword Block Message Header

MH_A64_OWDB - A64 Dual Oword Block Message Header		
Source:	DataPort 1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0-1	63:0	BlockOffset0
		Format: U64
		Specifies the U64 byte offset of Oword Block 0.
		Programming Notes
		If the BlockOffset is not in the 48-bit canonical address range, the access is Out-of-Bounds.
		Restriction
		The byte offset must be aligned to the message's data type. Dwords have [1:0] = 0, Qwords have [2:0] = 0, and Hwords have [4:0] = 0.
2-3	63:0	BlockOffset1
		Format: U64
		Specifies the U64 byte offset of Oword Block 1.
		Programming Notes
		If the BlockOffset is not in the 48-bit canonical address range, the access is Out-of-Bounds.
		Restriction
		The byte offset must be aligned to the message's data type. Dwords have [1:0] = 0, Qwords have [2:0] = 0, and Hwords have [4:0] = 0.
4-7	127:0	Reserved
		Format: Ignore Ignored

A64 Hword Block Message Header

MH_A64_HWB - A64 Hword Block Message Header		
Source:	DataPort 1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0-1	63:0	BlockOffset
		Format: U64
		Specifies the U64 byte offset of Oword block.
		Programming Notes
		If the BlockOffset is not in the 48-bit canonical address range, the access is Out-of-Bounds.
		Restriction
		The byte offset must be aligned to the message's data type. Dwords have [1:0] = 0, Qwords have [2:0] = 0, and Hwords have [4:0] = 0.
2-4	95:0	Reserved
		Format: Ignore
		Ignored
5	31:0	Hword Channel Mode
		Format: MHC_A64_CMODE
		Specifies the Hword Channel Mode
6-7	63:0	Reserved
		Format: Ignore
		Ignored

A64 Hword Data Blocks Message Descriptor Control Field

MDC_A64_DB_HW - A64 Hword Data Blocks Message Descriptor Control Field																						
Source:	BSpec																					
Size (in bits):	3																					
Default Value:	0x00000001																					
DWord	Bit	Description																				
0	2:0	<p>Data Blocks</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> </table> <p>Specifies the number of Hwords to be read or written</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 45%;">Name</th> <th style="width: 30%;">Description</th> </tr> </thead> <tbody> <tr> <td>01h</td> <td>HW1 [Default]</td> <td>1 Hword block</td> </tr> <tr> <td>02h</td> <td>HW2</td> <td>2 Hword blocks</td> </tr> <tr> <td>03h</td> <td>HW4</td> <td>4 Hword blocks</td> </tr> <tr> <td>04h</td> <td>HW8</td> <td>8 Hword blocks</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Format:	Enumeration	Value	Name	Description	01h	HW1 [Default]	1 Hword block	02h	HW2	2 Hword blocks	03h	HW4	4 Hword blocks	04h	HW8	8 Hword blocks	Others	Reserved	Ignored
Format:	Enumeration																					
Value	Name	Description																				
01h	HW1 [Default]	1 Hword block																				
02h	HW2	2 Hword blocks																				
03h	HW4	4 Hword blocks																				
04h	HW8	8 Hword blocks																				
Others	Reserved	Ignored																				

A64 Oword Block Message Header

MH_A64_OWB - A64 Oword Block Message Header		
Source:	DataPort 1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0-1	63:0	BlockOffset
		Format: U64
		Specifies the U64 byte offset of Oword block.
		Programming Notes
		If the BlockOffset is not in the 48-bit canonical address range, the access is Out-of-Bounds.
		Restriction
		The byte offset must be aligned to the message's data type. Dwords have [1:0] = 0, Qwords have [2:0] = 0, and Hwords have [4:0] = 0.
2-7	191:0	Reserved
		Format: Ignore
		Ignored

A64 Oword Data Blocks Message Descriptor Control Field

MDC_A64_DB_OW - A64 Oword Data Blocks Message Descriptor Control Field																											
Source:	BSpec																										
Size (in bits):	3																										
Default Value:	0x00000000																										
DWord	Bit	Description																									
0	2:0	<p>Data Blocks</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> <tr> <td colspan="2">Specifies the number of Oword blocks to be read or written</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">OW1L</td> <td>1 Oword, read into or written from the low 128 bits of the destination register</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">OW1U</td> <td>1 Oword, read into or written from the high 128 bits of the destination register</td> </tr> <tr> <td style="text-align: center;">02h</td> <td style="text-align: center;">OW2</td> <td>2 Owords</td> </tr> <tr> <td style="text-align: center;">03h</td> <td style="text-align: center;">OW4</td> <td>4 Owords</td> </tr> <tr> <td style="text-align: center;">04h</td> <td style="text-align: center;">OW8</td> <td>8 Owords</td> </tr> <tr> <td style="text-align: center;">Others</td> <td style="text-align: center;">Reserved</td> <td>Ignored</td> </tr> </table>	Format:	Enumeration	Specifies the number of Oword blocks to be read or written		Value	Name	Description	00h	OW1L	1 Oword, read into or written from the low 128 bits of the destination register	01h	OW1U	1 Oword, read into or written from the high 128 bits of the destination register	02h	OW2	2 Owords	03h	OW4	4 Owords	04h	OW8	8 Owords	Others	Reserved	Ignored
Format:	Enumeration																										
Specifies the number of Oword blocks to be read or written																											
Value	Name	Description																									
00h	OW1L	1 Oword, read into or written from the low 128 bits of the destination register																									
01h	OW1U	1 Oword, read into or written from the high 128 bits of the destination register																									
02h	OW2	2 Owords																									
03h	OW4	4 Owords																									
04h	OW8	8 Owords																									
Others	Reserved	Ignored																									

A64 Oword Dual Data Blocks Message Descriptor Control Field

MDC_A64_DB_OWD - A64 Oword Dual Data Blocks Message Descriptor Control Field																		
Source:	BSpec																	
Size (in bits):	3																	
Default Value:	0x00000001																	
DWord	Bit	Description																
0	2:0	<p>Data Blocks</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> <tr> <td colspan="2">Specifies the number of Oword blocks to be read or written</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">OWD1 [Default]</td> <td>1 Hword register, 2 Owords</td> </tr> <tr> <td style="text-align: center;">03h</td> <td style="text-align: center;">OWD4</td> <td>4 Hword registers, 8 Owords</td> </tr> <tr> <td style="text-align: center;">Others</td> <td style="text-align: center;">Reserved</td> <td>Ignored</td> </tr> </table>	Format:	Enumeration	Specifies the number of Oword blocks to be read or written		Value	Name	Description	01h	OWD1 [Default]	1 Hword register, 2 Owords	03h	OWD4	4 Hword registers, 8 Owords	Others	Reserved	Ignored
Format:	Enumeration																	
Specifies the number of Oword blocks to be read or written																		
Value	Name	Description																
01h	OWD1 [Default]	1 Hword register, 2 Owords																
03h	OWD4	4 Hword registers, 8 Owords																
Others	Reserved	Ignored																

A64 Scaled Data Port 2 Message Header

MH2_A64 - A64 Scaled Data Port 2 Message Header										
Source:	DataPort 2									
Size (in bits):	256									
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000									
DWord	Bit	Description								
0	31:0	<p>Surface Pitch</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MHC_PITCH</td> </tr> </table> <p>Specifies the 16-bit surface pitch to use with scaled A64 stateless messages.</p>	Format:	MHC_PITCH						
Format:	MHC_PITCH									
1-3	95:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore						
Format:	Ignore									
4-5	63:0	<p>Buffer Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">GraphicsAddress[63:0]</td> </tr> </table> <p>Specifies the surface base address[63:0] for scaled A64 stateless messages. The address must be Dword aligned (bits [1:0] = 0).</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">If the Buffer Base Address is not in 48-bit canonical form, the access is Out-of-Bounds.</td> </tr> <tr> <td colspan="2">Stateless A64 bounds checking rules specify that any address calculation that crosses the canonical address boundary is treated as out-of-bounds. A64 address calculations always use positive offsets, and do not support programming language constructs with negative offsets or indices (e.g. array[-1]).</td> </tr> </table>	Format:	GraphicsAddress[63:0]	Programming Notes		If the Buffer Base Address is not in 48-bit canonical form, the access is Out-of-Bounds.		Stateless A64 bounds checking rules specify that any address calculation that crosses the canonical address boundary is treated as out-of-bounds. A64 address calculations always use positive offsets, and do not support programming language constructs with negative offsets or indices (e.g. array[-1]).	
Format:	GraphicsAddress[63:0]									
Programming Notes										
If the Buffer Base Address is not in 48-bit canonical form, the access is Out-of-Bounds.										
Stateless A64 bounds checking rules specify that any address calculation that crosses the canonical address boundary is treated as out-of-bounds. A64 address calculations always use positive offsets, and do not support programming language constructs with negative offsets or indices (e.g. array[-1]).										
6-7	63:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore						
Format:	Ignore									

A64 Scaled Header Present Message Descriptor Control Field

MDC_A64_MHP - A64 Scaled Header Present Message Descriptor Control Field													
Source:	BSpec												
Size (in bits):	1												
Default Value:	0x00000000												
DWord	Bit	Description											
0	0	<p>Message Header Present</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> </table> <p>Specifies if the message uses the optional message header to modify the A64 address calculation, in combination with MDC_A64_SSO field.</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>No</td> <td>Message header is not present</td> </tr> <tr> <td>1h</td> <td>Yes</td> <td>Message header is present</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>The access is Out-of-Bounds if the SideBand Offset is enabled when the Message Header is not present.</p>	Format:	Enumeration	Value	Name	Description	0h	No	Message header is not present	1h	Yes	Message header is present
Format:	Enumeration												
Value	Name	Description											
0h	No	Message header is not present											
1h	Yes	Message header is present											

A64 Sideband Scale and Offset Enable Message Descriptor Control Field

MDC_A64_SBSO - A64 Sideband Scale and Offset Enable Message Descriptor Control Field								
Source:	BSpec							
Size (in bits):	8							
Default Value:	0x00000000							
DWord	Bit	Description						
0	7	<p>Sideband Offset Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If set, the Sideband Offset is added to the Message Header's Base Address Offset. The 16-bit Sideband Offset is specified in the extended function control field in the SEND instruction.</p> <table border="1"> <tr> <th colspan="2">Programming Notes</th> </tr> <tr> <td colspan="2">It is an illegal A64 encoding to enable the SideBand Offset when the Message Header is not present. That encoding combination is an SLM access in A32 message, but in an A64 message the illegal access is handled as Out-of-Bounds.</td> </tr> </table>	Format:	Enable	Programming Notes		It is an illegal A64 encoding to enable the SideBand Offset when the Message Header is not present. That encoding combination is an SLM access in A32 message, but in an A64 message the illegal access is handled as Out-of-Bounds.	
	Format:	Enable						
Programming Notes								
It is an illegal A64 encoding to enable the SideBand Offset when the Message Header is not present. That encoding combination is an SLM access in A32 message, but in an A64 message the illegal access is handled as Out-of-Bounds.								
6:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ					
Format:	MBZ							

AddrSubRegNum

AddrSubRegNum						
Source:	Eulsa					
Size (in bits):	4					
Default Value:	0x00000000					
<p>Address Subregister Number This field provides the subregister number for the address register. The address register contains 8 sub-registers. The size of each subregister is one word. The address register contains the register address of the operand, when the operand is in register-indirect addressing mode. This field applies to the destination operand and the source operands. It is ignored (or not present in the instruction word) for an immediate source operand. This field is present if the operand is in register-indirect addressing mode; it is not present if the operand is directly addressed. An address subregister used for indirect addressing is often called an index register.</p>						
DWord	Bit	Description				
0	3:0	<p>Address Subregister Number</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;">0-15</td> <td>Address Subregister Number</td> </tr> </tbody> </table>	Value	Name	0-15	Address Subregister Number
Value	Name					
0-15	Address Subregister Number					

Any Binding Table Index Message Descriptor Control Field

MDC_BTS_SLM_A32 - Any Binding Table Index Message Descriptor Control Field																														
Source:	BSpec																													
Size (in bits):	8																													
Default Value:	0x00000000																													
DWord	Bit	Description																												
0	7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>Enumeration</td> </tr> <tr> <td colspan="2">Specifies the surface for the message, which can be Surface State Model, SLM or Stateless.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td>00h-0EFh</td> <td>BTS</td> <td>Index of Binding Table State Surfaces</td> </tr> <tr> <td>F0h-0FBh</td> <td>Reserved</td> <td>Reserved for future use</td> </tr> <tr> <td>0FCh</td> <td>Reserved</td> <td>Reserved for future use</td> </tr> <tr> <td>0FCh</td> <td>SSO</td> <td>Specifies a Surface State Offset supplied by the extended message descriptor</td> </tr> <tr> <td>0FEh</td> <td>SLM</td> <td>Specifies an SLM access</td> </tr> <tr> <td>0FFh</td> <td>A32_A64</td> <td>Specifies a A32 or A64 Stateless access that is locally coherent (coherent within a thread group)</td> </tr> <tr> <td>0FDh</td> <td>A32_A64_NC</td> <td>Specifies a A32 or A64 Stateless access that is non-coherent (coherent within a thread).</td> </tr> </table> <p style="text-align: center;">Restriction</p> <p>When using A32_A64_NC, SW must ensure that 2 threads do not both access the same cache line (64B)</p>	Format:	Enumeration	Specifies the surface for the message, which can be Surface State Model, SLM or Stateless.		Value	Name	Description	00h-0EFh	BTS	Index of Binding Table State Surfaces	F0h-0FBh	Reserved	Reserved for future use	0FCh	Reserved	Reserved for future use	0FCh	SSO	Specifies a Surface State Offset supplied by the extended message descriptor	0FEh	SLM	Specifies an SLM access	0FFh	A32_A64	Specifies a A32 or A64 Stateless access that is locally coherent (coherent within a thread group)	0FDh	A32_A64_NC	Specifies a A32 or A64 Stateless access that is non-coherent (coherent within a thread).
Format:	Enumeration																													
Specifies the surface for the message, which can be Surface State Model, SLM or Stateless.																														
Value	Name	Description																												
00h-0EFh	BTS	Index of Binding Table State Surfaces																												
F0h-0FBh	Reserved	Reserved for future use																												
0FCh	Reserved	Reserved for future use																												
0FCh	SSO	Specifies a Surface State Offset supplied by the extended message descriptor																												
0FEh	SLM	Specifies an SLM access																												
0FFh	A32_A64	Specifies a A32 or A64 Stateless access that is locally coherent (coherent within a thread group)																												
0FDh	A32_A64_NC	Specifies a A32 or A64 Stateless access that is non-coherent (coherent within a thread).																												

Atomic Float Binary Operation Message Descriptor Control Field

MDC_FOP2 - Atomic Float Binary Operation Message Descriptor Control Field			
Source:	BSpec		
Size (in bits):	2		
Default Value:	0x00000001		
DWord	Bit	Description	
0	1:0	Atomic Float Operation Type	
		Format:	Enumeration
Specifies the atomic float binary operation to be performed			
Value	Name	Description	Programming Notes
01h	AOP_FMAX [Default]	new_dst = fmax(old_dst, src0)	The fmax operation implements the IEEE specification, which differs slightly from the DX specification when a source operand is a sNaN. fmax(x,qNaN) = fmax(qNaN,x) = x fmax(x,sNaN) = fmax(sNaN,x) = quietize(sNaN) fmax(sNaN,sNaN) = fmax(sNaN,qNaN) = fmax(qNaN,sNaN) = quietize(sNaN) fmax(qNaN,qNaN) = qNaN Fmax with sNaN operand returns sNaN instead of quietize(sNaN) Fmax(-0,+0) = -0. Should be +0, to match EU Fmax instruction.
02h	AOP_FMIN	new_dst = fmin(old_dst, src0)	The fmin operation implements the IEEE specification, which differs slightly from the DX specification when a source operand is a sNaN. fmin(x,qNaN) = fmin(qNaN,x) = x fmin(x,sNaN) = fmin(sNaN,x) = quietize(sNaN) fmin(sNaN,sNaN) = fmin(sNaN,qNaN) = fmin(qNaN,sNaN) = quietize(sNaN) fmin(qNaN,qNaN) = qNaN Fmin with sNaN operand returns sNaN instead of quietize(sNaN) Fmin(+0,-0) = +0. Should be -0, to match EU Fmin instruction.
Others	Reserved	Ignored	
Programming Notes			
When Return Data Control is set, old_dst is returned.			

Atomic Float Trinary Operation Message Descriptor Control Field

MDC_FOP3 - Atomic Float Trinary Operation Message Descriptor Control Field			
Source:	BSpec		
Size (in bits):	2		
Default Value:	0x00000003		
DWord	Bit	Description	
0	1:0	Atomic Float Operation Type Format: Enumeration Specifies the atomic float trinary operation to be performed	
		Value	Name
		Description	Programming Notes
		03h	AOP_FCMPWR [Default]
		new_dst = (src0 == old_dst) ? src1 : old_dst	The fcmpwr operation performs the comparison using IEEE specification rules, and performs the store as a raw move (so SNaN is not quietized). fcmpwr(NaN,x,y) = NaN fcmpwr(x, NaN,y) = x fcmpwr(x,x, NaN) = NaN
		Others	Reserved Ignored
		Programming Notes	
		When Return Data Control is set, old_dst is returned.	

Atomic Integer Binary Operation Message Descriptor Control Field

MDC_AOP2 - Atomic Integer Binary Operation Message Descriptor Control Field																																															
Source:	BSpec																																														
Size (in bits):	4																																														
Default Value:	0x00000001																																														
DWord	Bit	Description																																													
0	3:0	<p>Atomic Integer Operation Type</p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">Enumeration</td> </tr> <tr> <td colspan="3">Specifies the atomic integer binary operation to be performed</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>01h</td> <td>AOP_AND [Default]</td> <td>$\text{new_dst} = \text{old_dst} \text{ AND } \text{src0}$</td> </tr> <tr> <td>02h</td> <td>AOP_OR</td> <td>$\text{new_dst} = \text{old_dst} \text{src0}$</td> </tr> <tr> <td>03h</td> <td>AOP_XOR</td> <td>$\text{new_dst} = \text{old_dst} \wedge \text{src0}$</td> </tr> <tr> <td>04h</td> <td>AOP_MOV</td> <td>$\text{new_dst} = \text{src0}$</td> </tr> <tr> <td>07h</td> <td>AOP_ADD</td> <td>$\text{new_dst} = \text{old_dst} + \text{src0}$</td> </tr> <tr> <td>08h</td> <td>AOP_SUB</td> <td>$\text{new_dst} = \text{old_dst} - \text{src0}$</td> </tr> <tr> <td>09h</td> <td>AOP_REVSUB</td> <td>$\text{new_dst} = \text{src0} - \text{old_dst}$</td> </tr> <tr> <td>0Ah</td> <td>AOP_IMAX</td> <td>$\text{new_dst} = \text{imax}(\text{old_dst}, \text{src0})$</td> </tr> <tr> <td>0Bh</td> <td>AOP_IMIN</td> <td>$\text{new_dst} = \text{imin}(\text{old_dst}, \text{src0})$</td> </tr> <tr> <td>0Ch</td> <td>AOP_UMAX</td> <td>$\text{new_dst} = \text{umax}(\text{old_dst}, \text{src0})$</td> </tr> <tr> <td>0Dh</td> <td>AOP_UMIN</td> <td>$\text{new_dst} = \text{umin}(\text{old_dst}, \text{src0})$</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </table> <p style="text-align: center;">Programming Notes</p> <p>When Return Data Control is set, old_dst is returned.</p>	Format:	Enumeration		Specifies the atomic integer binary operation to be performed			Value	Name	Description	01h	AOP_AND [Default]	$\text{new_dst} = \text{old_dst} \text{ AND } \text{src0}$	02h	AOP_OR	$\text{new_dst} = \text{old_dst} \text{src0}$	03h	AOP_XOR	$\text{new_dst} = \text{old_dst} \wedge \text{src0}$	04h	AOP_MOV	$\text{new_dst} = \text{src0}$	07h	AOP_ADD	$\text{new_dst} = \text{old_dst} + \text{src0}$	08h	AOP_SUB	$\text{new_dst} = \text{old_dst} - \text{src0}$	09h	AOP_REVSUB	$\text{new_dst} = \text{src0} - \text{old_dst}$	0Ah	AOP_IMAX	$\text{new_dst} = \text{imax}(\text{old_dst}, \text{src0})$	0Bh	AOP_IMIN	$\text{new_dst} = \text{imin}(\text{old_dst}, \text{src0})$	0Ch	AOP_UMAX	$\text{new_dst} = \text{umax}(\text{old_dst}, \text{src0})$	0Dh	AOP_UMIN	$\text{new_dst} = \text{umin}(\text{old_dst}, \text{src0})$	Others	Reserved	Ignored
Format:	Enumeration																																														
Specifies the atomic integer binary operation to be performed																																															
Value	Name	Description																																													
01h	AOP_AND [Default]	$\text{new_dst} = \text{old_dst} \text{ AND } \text{src0}$																																													
02h	AOP_OR	$\text{new_dst} = \text{old_dst} \text{src0}$																																													
03h	AOP_XOR	$\text{new_dst} = \text{old_dst} \wedge \text{src0}$																																													
04h	AOP_MOV	$\text{new_dst} = \text{src0}$																																													
07h	AOP_ADD	$\text{new_dst} = \text{old_dst} + \text{src0}$																																													
08h	AOP_SUB	$\text{new_dst} = \text{old_dst} - \text{src0}$																																													
09h	AOP_REVSUB	$\text{new_dst} = \text{src0} - \text{old_dst}$																																													
0Ah	AOP_IMAX	$\text{new_dst} = \text{imax}(\text{old_dst}, \text{src0})$																																													
0Bh	AOP_IMIN	$\text{new_dst} = \text{imin}(\text{old_dst}, \text{src0})$																																													
0Ch	AOP_UMAX	$\text{new_dst} = \text{umax}(\text{old_dst}, \text{src0})$																																													
0Dh	AOP_UMIN	$\text{new_dst} = \text{umin}(\text{old_dst}, \text{src0})$																																													
Others	Reserved	Ignored																																													

Atomic Integer Trinary Operation Message Descriptor Control Field

MDC_AOP3 - Atomic Integer Trinary Operation Message Descriptor Control Field																				
Source:	BSpec																			
Size (in bits):	4																			
Default Value:	0x0000000E																			
DWord	Bit	Description																		
0	3:0	<p>Atomic Integer Operation Type</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td colspan="2">Enumeration</td> </tr> <tr> <td colspan="3">Specifies the atomic integer trinary operation to be performed</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">AOP_CMPWR_2W</td> <td style="text-align: center;">new_dst = (src0_2W == old_dst_2W) ? src1_2W : old_dst_2W</td> </tr> <tr> <td style="text-align: center;">0Eh</td> <td style="text-align: center;">AOP_CMPWR [Default]</td> <td style="text-align: center;">new_dst = (src0 == old_dst) ? src1 : old_dst</td> </tr> <tr> <td style="text-align: center;">Others</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Ignored</td> </tr> </table> <p style="text-align: center;">Programming Notes</p> <p>When Return Data Control is set, old_dst is returned.</p>	Format:	Enumeration		Specifies the atomic integer trinary operation to be performed			Value	Name	Description	00h	AOP_CMPWR_2W	new_dst = (src0_2W == old_dst_2W) ? src1_2W : old_dst_2W	0Eh	AOP_CMPWR [Default]	new_dst = (src0 == old_dst) ? src1 : old_dst	Others	Reserved	Ignored
Format:	Enumeration																			
Specifies the atomic integer trinary operation to be performed																				
Value	Name	Description																		
00h	AOP_CMPWR_2W	new_dst = (src0_2W == old_dst_2W) ? src1_2W : old_dst_2W																		
0Eh	AOP_CMPWR [Default]	new_dst = (src0 == old_dst) ? src1 : old_dst																		
Others	Reserved	Ignored																		

Atomic Integer Unary Operation Message Descriptor Control Field

MDC_AOP1 - Atomic Integer Unary Operation Message Descriptor Control Field																					
Source:	BSpec																				
Size (in bits):	4																				
Default Value:	0x00000005																				
DWord	Bit	Description																			
0	3:0	<p>Atomic Integer Operation Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enumeration</td> </tr> <tr> <td colspan="2">Specifies the atomic integer unary operation to be performed</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td>05h</td> <td>AOP_INC [Default]</td> <td>new_dst = old_dst + 1</td> </tr> <tr> <td>06h</td> <td>AOP_DEC</td> <td>new_dst = old_dst - 1</td> </tr> <tr> <td>0Fh</td> <td>AOP_PREDEC</td> <td>new_dst = old_dst - 1</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </table> <p style="text-align: center;">Programming Notes</p> <p>When Return Data Control is set, new_dst is returned by AOP_PREDEC and otherwise old_dst is returned.</p>	Format:	Enumeration	Specifies the atomic integer unary operation to be performed		Value	Name	Description	05h	AOP_INC [Default]	new_dst = old_dst + 1	06h	AOP_DEC	new_dst = old_dst - 1	0Fh	AOP_PREDEC	new_dst = old_dst - 1	Others	Reserved	Ignored
Format:	Enumeration																				
Specifies the atomic integer unary operation to be performed																					
Value	Name	Description																			
05h	AOP_INC [Default]	new_dst = old_dst + 1																			
06h	AOP_DEC	new_dst = old_dst - 1																			
0Fh	AOP_PREDEC	new_dst = old_dst - 1																			
Others	Reserved	Ignored																			



Audio Power State Format

Audio Power State Format														
Source:	BSpec													
Size (in bits):	2													
Default Value:	0x00000003													
DWord	Bit	Description												
0	1:0	Power State												
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>D0</td> <td>D0</td> </tr> <tr> <td>01b,10b</td> <td>Unsupported</td> <td>Unsupported</td> </tr> <tr> <td>11b</td> <td>D3 [Default]</td> <td>D3</td> </tr> </tbody> </table>	Value	Name	Description	00b	D0	D0	01b,10b	Unsupported	Unsupported	11b	D3 [Default]	D3
		Value	Name	Description										
		00b	D0	D0										
01b,10b	Unsupported	Unsupported												
11b	D3 [Default]	D3												

AVC CABAC

AVC CABAC		
Source:	VideoCS	
Size (in bits):	16	
Default Value:	0x00000000	
DWord	Bit	Description
0	15	Reserved Format: MBZ
	14	Coefficient level out-of-bound Error This flag indicates the coded coefficient level SEs in the bit-stream is out-of-bound.
	13	Reserved Format: MBZ
	12	Reserved Format: MBZ
	11	Temporal Direction Motion Vector Out-of-Bound Error This flag indicates motion vectors calculated from Temporal Direct Motion Vector is larger than the allowed range specified by the AVC spec.
	10	Reserved MBZ
	9	Motion Vector Delta SE Out-of-Bound Error This flag indicates inconsistent Motion Vector Delta SEs coded in the bit-stream.
	8	Reference Index SE Out-of-Bound Error This flag indicates inconsistent Reference Index SEs coded in the bit-stream.
	7	MacroBlock QpDelta Error This flag indicates out-of-bound MB QP delta SEs coded in the bit-stream.
	6	Motion Vector Delta SE Error This flag indicates out-of-bound motion vector delta SEs coded in the bit-stream.
	5	Reference Index SE Error This flag indicates out-of-bound Refidx SEs coded in the bit-stream.
	4	Residual Error This flag indicates out-of-bound absolute coefficient level SEs coded in the bit-stream.
	3	Slice end Error This flag indicates a pre-matured slice_end SE or inconsistent slice end on the last MB of a slice.
	2	Chroma Intra prediction Mode Error This flag indicates inconsistent Chroma Intra prediction mode SEs coded in the bit-stream.
	1	Luma Intra prediction Mode Error This flag indicates inconsistent luma Intra prediction mode SE coded in the bit-stream.
	0	MB Concealment Flag Each pulse from this flag indicates one MB is concealed by hardware.

AVC CAVLC

AVC CAVLC		
Source:	VideoCS	
Size (in bits):	16	
Default Value:	0x00000000	
DWord	Bit	Description
0	15	Total Zero out-of-bound Error This flag indicates the Total zero SE count exceed the max number of coeffs allowed in an intra16x16 AC block.
	14	Coefficient level out-of-bound Error This flag indicates the coded coefficient level SEs in the bit-stream is out-of-bound.
	13	RunBefore out-of-bound Error This flag indicates the coded RunBefore SE value is larger than the remaining zero block count.
	12	Total coefficient Out-of-bound Error This flag indicates the coded total coeff SE count exceed the max number of coeffs allowed in an intra16x16 AC block.
	11	Temporal Direction Motion Vector Out-of-Bound Error This flag indicates motion vectors calculated from Temporal Direct Motion Vector is larger than the allowed range specified by the AVC spec.
	10	Reserved Reserved
	9	Motion Vector Delta SE Out-of-Bound Error This flag indicates inconsistent Motion Vector Delta SEs coded in the bit-stream.
	8	Reference Index SE Out-of-Bound Error This flag indicates inconsistent Reference Index SEs coded in the bit-stream.
	7	RunBefore/TotalZero Error This flag indicates one or more inconsistent RunBefore or TotalZero SEs coded in the bit-stream.
	6	Exponential Golomb Error This flag indicates hardware detects more than 18 leadzero for skip and more than 19 for other SEs from the Exponential Golomb Logic
	5	Total Coeff SE Error This flag indicates one or more inconsistent total coeff SEs coded in the bit-stream.
	4	Macroblock Coded Block Pattern Error This flag indicates inconsistent CBP SEs coded in the bit-stream.
	3	Mbtype/submbtype Error This flag indicates inconsistent MBtype/SubMBtype SEs coded in the bit-stream.
2	Chroma Intra prediction Mode Error This flag indicates inconsistent Chroma Intra prediction mode SEs coded in the bit-stream.	
1	Luma Intra prediction Mode Error This flag indicates inconsistent luma Intra prediction mode SE coded in the bit-stream.	

AVC CAVLC

	0	MB Concealment Flag Each pulse from this flag indicates one MB is concealed by hardware.
--	---	--

AVS_Inline_DMEM		
8	31:0	Reserved
		Format: MBZ
9	31:0	Reserved
		Format: MBZ
10	31:0	Reserved
		Format: MBZ
11	31:0	Reserved
		Format: MBZ
12	31:0	Reserved
		Format: MBZ
13	31:0	Reserved
		Format: MBZ
14	31:0	Reserved
		Format: MBZ
15	31:0	Reserved
		Format: MBZ
16	31:0	Reserved
		Format: MBZ
17	31:0	Reserved
		Format: MBZ
18	31:1	Reserved
	0	hw_fixed_picture_qp AVS SE: fixed_picture_qp
19	31:6	Reserved
	5:0	hw_picture_qp AVS SE: picture_qp
20	31:1	Reserved
	0	hw_picture_reference_flag AVS SE: picture_reference_flag
21	31:1	Reserved
	0	hw_skip_mode_flag AVS SE: skip_mode_flag

AVS_Inline_DMEM		
22	31:1	Reserved Format: MBZ
	0	hw_loop_filter_diabie AVS SE: loop_filter_disable
23	31:0	Reserved Format: MBZ

BaseAddress4KByteAligned

BaseAddress4KByteAligned		
Source:	BSpec	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
Specifies a 64-bit (48-bit canonical) 4K-byte aligned memory base address.		
DWord	Bit	Description
0	31:12	Base Address Low Format: GraphicsAddress[31:12]
	11:0	Reserved Format: MBZ
1	31:16	Reserved Format: MBZ
	15:0	Base Address High Format: GraphicsAddress[47:32]

BCS Hardware-Detected Error Bit Definitions

BCS Hardware-Detected Error Bit Definitions											
Source:	BlitterCS										
Size (in bits):	16										
Default Value:	0x00000000										
DWord	Bit	Description									
0	15:3	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	2	Command Privilege Violation Error This bit is set if a command classified as privileged is parsed in a non-privileged batch buffer. The command will be converted to a NOOP and parsing will continue.									
	1	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ										
0	Instruction Error This bit is set when the Renderer Instruction Parser detects an error while parsing an instruction. Instruction errors include: <ul style="list-style-type: none"> Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported). Defeated MI Instruction Opcodes: <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> <td>Instruction Error detected</td> </tr> </tbody> </table> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th colspan="2">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">This error indications cannot be cleared except by reset (i.e., it is a fatal error).</td> </tr> </tbody> </table>	Value	Name	Description	1		Instruction Error detected	Programming Notes		This error indications cannot be cleared except by reset (i.e., it is a fatal error).	
Value	Name	Description									
1		Instruction Error detected									
Programming Notes											
This error indications cannot be cleared except by reset (i.e., it is a fatal error).											

BINDING_TABLE_EDIT_ENTRY

BINDING_TABLE_EDIT_ENTRY				
Source:	RenderCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:24	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	23:16	<p>Binding Table Index</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>This field specifies the index of binding table entry that will be updated.</p>	Format:	U8
Format:	U8			
15:0	<p>Surface State Pointer</p> <table border="1"> <tr> <td>Format:</td> <td>SurfaceStateOffset[21:6]RENDER_SURFACE_STATE</td> </tr> </table> <p>Surface State Pointer. This address points to a surface state block. This pointer is relative to the Surface State Base Address.</p>	Format:	SurfaceStateOffset[21:6]RENDER_SURFACE_STATE	
Format:	SurfaceStateOffset[21:6]RENDER_SURFACE_STATE			

BINDING_TABLE_STATE

BINDING_TABLE_STATE				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
<p>The binding table binds surfaces to logical resource indices used by shaders and other compute engine kernels. It is stored as an array of up to 256 elements, each of which contains one dword as defined here. The start of each element is spaced one dword apart. The first element of the binding table is aligned to a 64-byte boundary. Binding table indexes beyond 256 will automatically be mapped to entry 0 by the HW, w/ the exception of any messages which support the special indexes 240 through 255, inclusive.</p>				
DWord	Bit	Description		
0	31:6	<p>Surface State Pointer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>SurfaceStateOffset[31:6]</td> </tr> </table> <p>This 64-byte aligned address points to a surface state block. This pointer is relative to the Surface State Base Address</p>	Format:	SurfaceStateOffset[31:6]
	Format:	SurfaceStateOffset[31:6]		
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			

Bit Definition for Interrupt Control Registers - Blitter

Bit Definition for Interrupt Control Registers - Blitter				
Source:	BlitterCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:28	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>These bits may be assigned to interrupts on future products/steppings.</p>	Format:	MBZ
	Format:	MBZ		
	27	<p>Wait on Semaphore</p> <p>Exec-List Scheduling: Set 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>		
	26:25	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	24	<p>Context Switch Interrupt</p> <p>Set when a context switch has just occurred. Exec-List Enable bit needs to be set for this interrupt to occur.</p>		
	23	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	22	<p>Timeout Counter Expired</p> <p>Set when the Blitter Command Streamer timeout counter has reached the timeout threshold value. This feature is not POR and SW must never unmask (enable) this interrupt via BCS_IMR.</p>		
	21	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
20	<p>MI_FLUSH_DW Notify Interrupt</p> <p>The Pipe Control packet (Fences) specified in 3D pipeline document may optionally generate an Interrupt. The Store QW associated with a fence is completed ahead of the interrupt.</p>			
19	<p>Blitter Command Parser Master Error</p> <p>When this status bit is set, it indicates that the hardware has detected an error. It is set by the device upon an error condition and cleared by a CPU write of a one to the appropriate bit contained in the Error ID register followed by a write of a one to this bit in the IIR. Further information on the source of the error comes from the "Error Status Register" which along with the "Error Mask Register" determine which error conditions will cause the error status bit to be set and the interrupt to occur.</p> <p>Page Table Error: Indicates a page table error.</p> <p>Instruction Parser Error: The Blitter Instruction Parser encounters an error while parsing an instruction.</p>			
18:17	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			

Bit Definition for Interrupt Control Registers - Blitter

16	<p>Blitter Command Parser User Interrupt</p> <p>This status bit is set when an MI_USER_INTERRUPT instruction is executed on the Blitter Command Parser. Note that instruction execution is not halted and proceeds normally. A mechanism such as an MI_STORE_DATA instruction is required to associate a particular meaning to a user interrupt.</p>			
15:0	<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:	MBZ
Format:	MBZ			

Bit Definition for Interrupt Control Registers - Media#1

Bit Definition for Interrupt Control Registers - Media#1			
Source:	VideoCS		
Size (in bits):	32		
Default Value:	0x00000000		
DWord	Bit	Description	
0	31:16	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td></tr></table> MBZ	
	15:12	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td></tr></table> MBZ These bits may be assigned to interrupts on future products/steppings.	
	11	Wait on Semaphore Exec-List Scheduling: Set 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.	
	10	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td></tr></table> MBZ	
	9	Reserved	
	8	Context Switch Interrupt Set when a context switch has just occurred. ExecList Enable bit needs to be set for this interrupt to occur.	
	7	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td></tr></table> MBZ	
	6	Timeout Counter Expired Set when the VCS timeout counter has reached the timeout thresh-hold value.	
5	Reserved		
4	MI_FLUSH_DW Notify Interrupt The Pipe Control packet (Fences) specified in 3D pipeline document may optionally generate an Interrupt. The Store QW associated with a fence is completed ahead of the interrupt.		
3	Video Command Parser Master Error When this status bit is set, it indicates that the hardware has detected an error. It is set by the device upon an error condition and cleared by a CPU write of a one to the appropriate bit contained in the Error ID register followed by a write of a one to this bit in the IIR. Further information on the source of the error comes from the "Error Status Register" which along with the "Error Mask Register" determine which error conditions will cause the error status bit to be set and the interrupt to occur. Page Table Error: Indicates a page table error. Instruction Parser Error: The Blitter Instruction Parser encounters an error while parsing an instruction.		

Bit Definition for Interrupt Control Registers - Media#1			
2:1	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
0	<p>Video Command Parser User Interrupt</p> <p>This status bit is set when an MI_USER_INTERRUPT instruction is executed on the Video Command Parser. Note that instruction execution is not halted and proceeds normally. A mechanism such as an MI_STORE_DATA instruction is required to associate a particular meaning to a user interrupt.</p>		

Bit Definition for Interrupt Control Registers - Media#2

Bit Definition for Interrupt Control Registers - Media#2				
Source:	VideoCS2			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:28	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>These bits may be assigned to interrupts on future products/steppings.</p>	Format:	MBZ
	Format:	MBZ		
	27	<p>Wait on Semaphore Exec-List Scheduling: Set 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>		
	26	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	25	Reserved		
	24	<p>Context Switch Interrupt Set when a context switch has just occurred. Execlist Enable bit needs to be set for this interrupt to occur.</p>		
	23	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	22	<p>Timeout Counter Expired Set when the VCS timeout counter has reached the timeout thresh-hold value.</p>		
	21	Reserved		
	20	<p>MI_FLUSH_DW Notify Interrupt The Pipe Control packet (Fences) specified in 3D pipeline document may optionally generate an Interrupt. The Store QW associated with a fence is completed ahead of the interrupt.</p>		
19	<p>Video Command Parser Master Error When this status bit is set, it indicates that the hardware has detected an error. It is set by the device upon an error condition and cleared by a CPU write of a one to the appropriate bit contained in the Error ID register followed by a write of a one to this bit in the IIR. Further information on the source of the error comes from the "Error Status Register" which along with the "Error Mask Register" determine which error conditions will cause the error status bit to be set and the interrupt to occur. Page Table Error: Indicates a page table error. Instruction Parser Error: The Blitter Instruction Parser encounters an error while parsing an instruction.</p>			
18:17	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			

Bit Definition for Interrupt Control Registers - Media#2

16	Video Command Parser User Interrupt This status bit is set when an MI_USER_INTERRUPT instruction is executed on the Video Command Parser. Note that instruction execution is not halted and proceeds normally. A mechanism such as an MI_STORE_DATA instruction is required to associate a particular meaning to a user interrupt.	
15:0	Reserved	
	Format:	MBZ

Bit Definition for Interrupt Control Registers - Render

Bit Definition for Interrupt Control Registers - Render						
Source:	RenderCS					
Size (in bits):	32					
Default Value:	0x00000000					
DWord	Bit	Description				
0	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Reserved for other command streamers - cannot be allocated by main command streamer.</p>	Format:	MBZ		
	Format:	MBZ				
	15:12	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	11	<p>Wait on Semaphore</p> <p>Exec-List Scheduling: Set 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>				
	10	<p>L3 Counter Save Interrupt</p>				
	9	<p>TR Invalid Tile Detection</p> <p>Hardware generates an interrupt when tiled resources translations hit an INVALID tile.</p>				
	8	<p>Context Switch Interrupt</p> <p>Set when a context switch has just occurred. Execlist Enable bit needs to be set for this interrupt to occur.</p>				
	7	<p>Page Fault</p> <table border="1"> <thead> <tr> <th colspan="2">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">This interrupt is for handling Legacy Page Fault interface for all Command Streamers (BCS, RCS, VCS, VECS). When Fault Repair Mode is enabled, Interrupt mask register value is not looked at to generate interrupt due to page fault. Please refer to vol1c "Page Fault Support" section for more details.</td> </tr> </tbody> </table>	Description		This interrupt is for handling Legacy Page Fault interface for all Command Streamers (BCS, RCS, VCS, VECS). When Fault Repair Mode is enabled, Interrupt mask register value is not looked at to generate interrupt due to page fault. Please refer to vol1c "Page Fault Support" section for more details.	
	Description					
This interrupt is for handling Legacy Page Fault interface for all Command Streamers (BCS, RCS, VCS, VECS). When Fault Repair Mode is enabled, Interrupt mask register value is not looked at to generate interrupt due to page fault. Please refer to vol1c "Page Fault Support" section for more details.						
6	<p>Timeout Counter Expired</p> <p>Set when the render pipe timeout counter (0x02190) has reached the timeout threshold value (0x0217c).</p>					
5	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
4	<p>PIPE_CONTROL Notify Interrupt</p> <p>The Pipe Control packet (Fences) specified in 3D pipeline document may optionally generate an Interrupt. The Store QW associated with a fence is completed ahead of the interrupt.</p>					

Bit Definition for Interrupt Control Registers - Render			
3	<p>Render Command Parser Master Error</p> <p>When this status bit is set, it indicates that the hardware has detected an error. It is set by the device upon an error condition and cleared by a CPU write of a one to the appropriate bit contained in the Error ID register followed by a write of a one to this bit in the IIR. Further information on the source of the error comes from the "Error Status Register" which along with the "Error Mask Register" determine which error conditions will cause the error status bit to be set and the interrupt to occur.</p> <p>Page Table Error: Indicates a page table error.</p> <p>Instruction Parser Error: The Render Instruction Parser encounters an error while parsing an instruction.</p>		
2	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%; padding: 2px;">Format:</td> <td style="width: 30%; padding: 2px;">MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ		
1	<p>Reserved</p>		
0	<p>Render Command Parser User Interrupt</p> <p>This status bit is set when an MI_USER_INTERRUPT instruction is executed on the Render Command Parser. Note that instruction execution is not halted and proceeds normally. A mechanism such as an MI_STORE_DATA instruction is required to associate a particular meaning to a user interrupt.</p>		

Bit Definition for Interrupt Control Registers - Video Enhancement

Bit Definition for Interrupt Control Registers - Video Enhancement				
Source:	VideoEnhancementCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:12	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>These bits may be assigned to interrupts on future products/steppings.</p>	Format:	MBZ
	Format:	MBZ		
	11	<p>Wait on Semaphore</p> <p>Exec-List Scheduling: Set 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>		
	10	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	9	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	8	<p>Context Switch Interrupt</p> <p>Set when a context switch has just occurred. Exec-List Enable bit needs to be set for this interrupt to occur.</p>		
	7	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
6	<p>Timeout Counter Expired</p> <p>Set when the Video Enhancement Command Streamer timeout counter has reached the timeout threshold value. This feature is not POR and SW must never unmask (enable) this interrupt via VECS_IMR.</p>			
5	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			
4	<p>MI_FLUSH_DW Notify Interrupt</p> <p>The Pipe Control packet (Fences) specified in 3D pipeline document may optionally generate an Interrupt. The Store QW associated with a fence is completed ahead of the interrupt.</p>			
3	<p>Video Enhancement Command Parser Master Error</p> <p>When this status bit is set, it indicates that the hardware has detected an error. It is set by the device upon an error condition and cleared by a CPU write of a one to the appropriate bit contained in the Error ID register followed by a write of a one to this bit in the IIR. Further information on the source of the error comes from the "Error Status Register" which along with the "Error Mask Register" determine which error conditions will cause the error status bit to be set and the interrupt to occur.</p> <p>Page Table Error: Indicates a page table error.</p> <p>Instruction Parser Error: The Blitter Instruction Parser encounters an error while parsing an instruction.</p>			

Bit Definition for Interrupt Control Registers - Video Enhancement

	2:1	Reserved	
		Format:	MBZ
	0	<p>Video Enhancement Command Parser User Interrupt</p> <p>This status bit is set when an MI_USER_INTERRUPT instruction is executed on the Video Enhancement Command Parser. Note that instruction execution is not halted and proceeds normally. A mechanism such as an MI_STORE_DATA instruction is required to associate a particular meaning to a user interrupt.</p>	

BLEND_STATE

BLEND_STATE				
Source:	BSpec			
Size (in bits):	544			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
<p>The blend state is stored as a structure containing a common DWORD that applies to all RTs and an array of up to 8 elements, each of which contains the two DWords for each. The start of each element is spaced 2 DWords apart. The blend state is aligned to a 64-byte boundary, which is pointed to by a field in 3DSTATE_BLEND_STATE_POINTERS. The 3-bit Render Target Index field in the Render Target Write data port message header is used to select which of the 8 elements from BLEND_STATE that is used on the current message.</p>				
DWord	Bit	Description		
0	31	<p>Alpha To Coverage Enable</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If set, Source0 Alpha is converted to a temporary 1/2/4-bit coverage mask and the mask bit corresponding to the sample# ANDed with the sample mask bit. If set, sample coverage is computed based on src0 alpha value. Value of 0 disables all samples and value of 1 enables all samples for that pixel. The same coverage needs to apply to all the RTs in MRT case. Further, any value of src0 alpha between 0 and 1 monotonically increases the number of enabled pixels. The field is applied to all the RTs in MRT case.</p>	Format:	Enable
	Format:	Enable		
	30	<p>Independent Alpha Blend Enable</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>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. The field is applied to all the RTs in MRT case.</p>	Format:	Enable
	Format:	Enable		
29	<p>Alpha To One Enable</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If set, Source0 Alpha is set to 1.0f after (possibly) being used to generate the AlphaToCoverage coverage mask. If Dual Source Blending is enabled, this bit must be disabled. The field is applied to all the RTs in MRT case.</p>	Format:	Enable	
Format:	Enable			
28	<p>Alpha To Coverage Dither Enable</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If set, sample coverage is computed based on src0 alpha value and it modulates the sample coverage based on screen coordinates. Value of 0 disables all samples and value of 1 enables all samples for that pixel. The same coverage needs to apply to all the RTs in MRT case. Further, any value of src0 alpha between 0 and 1 monotonically increases the number of enabled pixels. If AlphaToCoverage is disabled, AlphaToCoverage Dither does not have any impact. The field is applied to all the RTs in MRT case.</p>	Format:	Enable	
Format:	Enable			

BLEND_STATE																	
1..16	<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">27</td> <td style="width: 15%;">Alpha Test Enable</td> <td style="width: 60%;">Format:</td> <td style="width: 20%; text-align: center;">Enable</td> </tr> <tr> <td colspan="4"> <p>Enables the AlphaTest function of the Pixel Processing pipeline. The field is applied to all the RTs in MRT case.</p> </td> </tr> <tr> <td colspan="4" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="4"> <p>Alpha Test can only be enabled if Pixel Shader outputs a float alpha value. Alpha Test is applied independently on each render target by comparing that render target's alpha value against the alpha reference value. If the alpha test fails, the corresponding pixel write will be suppressed only for that render target. The depth/stencil update will occur if alpha test passes for any render target.</p> </td> </tr> </table>	27	Alpha Test Enable	Format:	Enable	<p>Enables the AlphaTest function of the Pixel Processing pipeline. The field is applied to all the RTs in MRT case.</p>				Programming Notes				<p>Alpha Test can only be enabled if Pixel Shader outputs a float alpha value. Alpha Test is applied independently on each render target by comparing that render target's alpha value against the alpha reference value. If the alpha test fails, the corresponding pixel write will be suppressed only for that render target. The depth/stencil update will occur if alpha test passes for any render target.</p>			
	27	Alpha Test Enable	Format:	Enable													
	<p>Enables the AlphaTest function of the Pixel Processing pipeline. The field is applied to all the RTs in MRT case.</p>																
	Programming Notes																
	<p>Alpha Test can only be enabled if Pixel Shader outputs a float alpha value. Alpha Test is applied independently on each render target by comparing that render target's alpha value against the alpha reference value. If the alpha test fails, the corresponding pixel write will be suppressed only for that render target. The depth/stencil update will occur if alpha test passes for any render target.</p>																
	<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">26:24</td> <td style="width: 15%;">Alpha Test Function</td> <td style="width: 60%;">Format:</td> <td style="width: 20%; text-align: center;">3D_Compare_Function</td> </tr> <tr> <td colspan="4"> <p>This field specifies the comparison function used in the AlphaTest function. The field is applied to all the RTs in MRT case.</p> </td> </tr> </table>	26:24	Alpha Test Function	Format:	3D_Compare_Function	<p>This field specifies the comparison function used in the AlphaTest function. The field is applied to all the RTs in MRT case.</p>											
	26:24	Alpha Test Function	Format:	3D_Compare_Function													
<p>This field specifies the comparison function used in the AlphaTest function. The field is applied to all the RTs in MRT case.</p>																	
<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">23</td> <td style="width: 15%;">Color Dither Enable</td> <td style="width: 60%;">Format:</td> <td style="width: 20%; text-align: center;">Enable</td> </tr> <tr> <td colspan="4"> <p>Enables dithering of colors (including any alpha component) before they are written to the Color Buffer. The field is applied to all the RTs in MRT case.</p> </td> </tr> <tr> <td colspan="4" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="4"> <p>For YUV render target formats, this field must be programmed to 0.</p> </td> </tr> </table>	23	Color Dither Enable	Format:	Enable	<p>Enables dithering of colors (including any alpha component) before they are written to the Color Buffer. The field is applied to all the RTs in MRT case.</p>				Programming Notes				<p>For YUV render target formats, this field must be programmed to 0.</p>				
23	Color Dither Enable	Format:	Enable														
<p>Enables dithering of colors (including any alpha component) before they are written to the Color Buffer. The field is applied to all the RTs in MRT case.</p>																	
Programming Notes																	
<p>For YUV render target formats, this field must be programmed to 0.</p>																	
<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">22:21</td> <td style="width: 15%;">X Dither Offset</td> <td style="width: 60%;">Format:</td> <td style="width: 20%; text-align: center;">U2</td> </tr> <tr> <td colspan="4"> <p>Specifies offset to apply to pixel X coordinate LSBs when accessing dither table. The field is applied to all the RTs in MRT case.</p> </td> </tr> </table>	22:21	X Dither Offset	Format:	U2	<p>Specifies offset to apply to pixel X coordinate LSBs when accessing dither table. The field is applied to all the RTs in MRT case.</p>												
22:21	X Dither Offset	Format:	U2														
<p>Specifies offset to apply to pixel X coordinate LSBs when accessing dither table. The field is applied to all the RTs in MRT case.</p>																	
<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">20:19</td> <td style="width: 15%;">Y Dither Offset</td> <td style="width: 60%;">Format:</td> <td style="width: 20%; text-align: center;">U2</td> </tr> <tr> <td colspan="4"> <p>Specifies offset to apply to pixel Y coordinate LSBs when accessing dither table. The field is applied to all the RTs in MRT case.</p> </td> </tr> </table>	20:19	Y Dither Offset	Format:	U2	<p>Specifies offset to apply to pixel Y coordinate LSBs when accessing dither table. The field is applied to all the RTs in MRT case.</p>												
20:19	Y Dither Offset	Format:	U2														
<p>Specifies offset to apply to pixel Y coordinate LSBs when accessing dither table. The field is applied to all the RTs in MRT case.</p>																	
<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">18:0</td> <td style="width: 15%;">Reserved</td> <td style="width: 60%;">Format:</td> <td style="width: 20%; text-align: center;">MBZ</td> </tr> </table>	18:0	Reserved	Format:	MBZ													
18:0	Reserved	Format:	MBZ														
<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">63:0</td> <td style="width: 15%;">Entry</td> <td style="width: 60%;">Format:</td> <td style="width: 20%; text-align: center;">BLEND_STATE_ENTRY</td> </tr> </table>	63:0	Entry	Format:	BLEND_STATE_ENTRY													
63:0	Entry	Format:	BLEND_STATE_ENTRY														

BLEND_STATE_ENTRY

BLEND_STATE_ENTRY				
Source:	BSpec			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0..1	63	Logic Op Enable		
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td>Enable</td></tr></table>		Enable
			Enable	
		Enables the LogicOp function of the Pixel Processing pipeline.		
Programming Notes				
		Enabling LogicOp and Color Buffer Blending at the same time is UNDEFINED		
62:59		Logic Op Function		
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td>3D_Logic_Op_Function</td></tr></table>		3D_Logic_Op_Function
	3D_Logic_Op_Function			
This field specifies the function to be performed (when enabled) in the Logic Op stage of the Pixel Processing pipeline. Note that the encoding of this field is one less than the corresponding "R2_" ROP code defined in WINGDI.H, and is a rather contorted mapping of the OpenGL LogicOp encodings. However, this field was defined such that, when the 4 bits are replicated to 8 bits, they coincide with the ROP codes used in the Blter. Note: if the Logic Op Function does not depend on "D", the dest buffer is not read.				
58:37		Reserved		
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td>MBZ</td></tr></table>		MBZ
	MBZ			
36		Pre-Blend Source Only Clamp Enable		
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td>Enable</td></tr></table>		Enable
			Enable	
		This field specifies whether the source(s) are clamped prior to blending, regardless of whether blending is enabled. If DISABLED, no clamping is performed prior to blending. If ENABLED, only source0 and source 1, if dual source is enabled, are clamped prior to the blend to the range specified by Color Clamp Range.		
		Value Name Description		
		0	Disabled	No clamping is performed prior to blending.
1	Enabled	Only Source(s) are clamped prior to blend function. Other inputs to blend must not be clamped.		
		Programming Notes		
See table in Pre-Blending Color Clamp subsection for programming restrictions as a function of RT format. This field is ignored (treated as DISABLED) for UINT and SINT RT surface formats. Blending is not supported for those RT surface formats. When this bit is enabled Pre-Blend Color Clamp Enable RT[0] must be disabled.				

BLEND_STATE_ENTRY																	
35:34	<p>Color Clamp Range Specifies the clamped range used in Pre-Blend and Post-Blend Color Clamp functions if one or both of those functions are enabled. Note that this range selection is shared between those functions. This field is ignored if both of the Color Clamp Enables are disabled</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>COLORCLAMP_UNORM</td> <td>Clamp Range [0,1]</td> </tr> <tr> <td>1</td> <td>COLORCLAMP_SNORM</td> <td>Clamp Range [-1,1]</td> </tr> <tr> <td>2</td> <td>COLORCLAMP_RTFORMAT</td> <td>Clamp to the range of the RT surface format (Note: The Alpha component is clamped to FLOAT16 for R11G11B10_FLOAT format).</td> </tr> <tr> <td>3</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>		Value	Name	Description	0	COLORCLAMP_UNORM	Clamp Range [0,1]	1	COLORCLAMP_SNORM	Clamp Range [-1,1]	2	COLORCLAMP_RTFORMAT	Clamp to the range of the RT surface format (Note: The Alpha component is clamped to FLOAT16 for R11G11B10_FLOAT format).	3	Reserved	Reserved
Value	Name	Description															
0	COLORCLAMP_UNORM	Clamp Range [0,1]															
1	COLORCLAMP_SNORM	Clamp Range [-1,1]															
2	COLORCLAMP_RTFORMAT	Clamp to the range of the RT surface format (Note: The Alpha component is clamped to FLOAT16 for R11G11B10_FLOAT format).															
3	Reserved	Reserved															
33	<p>Pre-Blend Color Clamp Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This field specifies whether the source, destination and constant color channels are clamped prior to blending, regardless of whether blending is enabled. If DISABLED, no clamping is performed prior to blending. If ENABLED, all inputs to the blend function are clamped prior to the blend to the range specified by Color Clamp Range.</p> <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>No clamping is performed prior to blending.</td> </tr> <tr> <td>1</td> <td>Enabled</td> <td>All inputs to the blend function are clamped prior to the blend to the range specified by Color Clamp Range.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>See table in Pre-Blending Color Clamp subsection for programming restrictions as a function of RT format. This field is ignored (treated as DISABLED) for UINT and SINT RT surface formats. Blending is not supported for those RT surface formats. The device will automatically clamp source color channels to the respective RT surface range.</p>		Format:	Enable	Value	Name	Description	0	Disabled	No clamping is performed prior to blending.	1	Enabled	All inputs to the blend function are clamped prior to the blend to the range specified by Color Clamp Range.				
Format:	Enable																
Value	Name	Description															
0	Disabled	No clamping is performed prior to blending.															
1	Enabled	All inputs to the blend function are clamped prior to the blend to the range specified by Color Clamp Range.															
32	<p>Post-Blend Color Clamp Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If blending is enabled, this field specifies whether the blending output channels are first clamped to the range specified by Color Clamp Range. Regardless of whether this clamping is enabled, the blending output channels will be clamped to the RT surface format just prior to being written.</p> <p style="text-align: center;">Programming Notes</p> <p>See table in Pre-Blending Color Clamp subsection for programming restrictions as a function of RT format. This field is ignored (treated as DISABLED) for UINT and SINT RT surface formats. Blending is not supported for those RT surface formats. The device will automatically clamp source color channels to the respective RT surface range. When this bit is enabled Pre-Blend Source Only Clamp Enable RT[0] must be disabled.</p>		Format:	Enable													
Format:	Enable																

BLEND_STATE_ENTRY				
31	<p>Color Buffer Blend Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Enables the ColorBufferBlending (nee "alpha blending") function of the Pixel Processing Pipeline for this render target.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>Enabling LogicOp and ColorBufferBlending at the same time is UNDEFINED</p>	Format:	Enable	Programming Notes
Format:	Enable			
Programming Notes				
30:26	<p>Source Blend Factor</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>3D_Color_Buffer_Blend_Factor</td> </tr> </table> <p>Controls the "source factor" in the ColorBufferBlending function. Refer to Source Alpha Blend Factor for encodings.</p>	Format:	3D_Color_Buffer_Blend_Factor	
Format:	3D_Color_Buffer_Blend_Factor			
25:21	<p>Destination Blend Factor</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>3D_Color_Buffer_Blend_Factor</td> </tr> </table> <p>Controls the "destination factor" in the ColorBufferBlending function. Refer to Source Alpha Blend Factor for encodings.</p>	Format:	3D_Color_Buffer_Blend_Factor	
Format:	3D_Color_Buffer_Blend_Factor			
20:18	<p>Color Blend Function</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>3D_Color_Buffer_Blend_Function</td> </tr> </table> <p>This field specifies the function used to combine the color components in the ColorBufferBlending function of the Pixel Processing Pipeline. If Independent Alpha Blend Enable is disabled, this field will also control the blending of the alpha components in the ColorBufferBlending function.</p>	Format:	3D_Color_Buffer_Blend_Function	
Format:	3D_Color_Buffer_Blend_Function			
17:13	<p>Source Alpha Blend Factor</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>3D_Color_Buffer_Blend_Factor</td> </tr> </table> <p>Controls the "source factor" in alpha Color Buffer Blending stage. Note: For the source/destination alpha blend factors, the encodings indicating "COLOR" are the same as the encodings indicating "ALPHA", as the alpha component of the color is selected.</p>	Format:	3D_Color_Buffer_Blend_Factor	
Format:	3D_Color_Buffer_Blend_Factor			
12:8	<p>Destination Alpha Blend Factor</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>3D_Color_Buffer_Blend_Factor</td> </tr> </table> <p>Controls the "destination factor" in alpha Color Buffer Blending stage. Refer to Source Alpha Blend Factor for encodings.</p>	Format:	3D_Color_Buffer_Blend_Factor	
Format:	3D_Color_Buffer_Blend_Factor			
7:5	<p>Alpha Blend Function</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 25%;">Format:</td> <td>3D_Color_Buffer_Blend_Function</td> </tr> </table> <p>This field specifies the function used to combine the alpha components in the Color Buffer blend stage of the Pixel Pipeline when the IndependentAlphaBlend state is enabled.</p>	Format:	3D_Color_Buffer_Blend_Function	
Format:	3D_Color_Buffer_Blend_Function			
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			

BLEND_STATE_ENTRY				
3	Write Disable Alpha			
	Format:	Disable		
	This field controls the writing of the alpha component into the Render Target.			
	Value	Name	Description	
	0b	Enabled	Alpha component can be overwritten	
	1b	Disabled	Writes to the color buffer will not modify Alpha.	
	Programming Notes			
	For YUV surfaces, this field must be set to 0B (enabled).			
	2	Write Disable Red		
		Format:	Disable	
		This field controls the writing of the red component into the Render Target.		
		Value	Name	Description
		0b	Enabled	Red component can be overwritten
		1b	Disabled	Writes to the color buffer will not modify Red.
		Programming Notes		
		For YUV surfaces, this field must be set to 0B (enabled).		
1		Write Disable Green		
		Format:	Disable	
	This field controls the writing of the green component into the Render Target.			
	Value	Name	Description	
	0b	Enabled	Green component can be overwritten	
	1b	Disabled	Writes to the color buffer will not modify Green.	
	Programming Notes			
	For YUV surfaces, this field must be set to 0B (enabled).			
	0	Write Disable Blue		
		Format:	Disable	
This field controls the writing of the Blue component into the Render Target.				
Value		Name	Description	
0b		Enabled	Blue component can be overwritten	
1b		Disabled	Writes to the color buffer will not modify Blue.	
Programming Notes				
For YUV surfaces, this field must be set to 0B (enabled).				

Block Dimensions Message Header Control

MHC_BDIM - Block Dimensions Message Header Control																	
Source:	BSpec																
Size (in bits):	32																
Default Value:	0x00000000																
DWord	Bit	Description															
0	31:22	Reserved															
		Format: Ignore Ignored															
	21:20	Block Height															
		Format: Enumeration Height in rows of block being accessed. Range = [0,3] representing 1 to 8 rows.															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>H1</td> <td>Block height = 1 row</td> </tr> <tr> <td>1h</td> <td>H2</td> <td>Block height = 2 rows</td> </tr> <tr> <td>2h</td> <td>H4</td> <td>Block height = 4 rows</td> </tr> <tr> <td>03h</td> <td>H8</td> <td>Block height = 8 rows</td> </tr> </tbody> </table>	Value	Name	Description	0h	H1	Block height = 1 row	1h	H2	Block height = 2 rows	2h	H4	Block height = 4 rows	03h	H8	Block height = 8 rows
		Value	Name	Description													
		0h	H1	Block height = 1 row													
		1h	H2	Block height = 2 rows													
		2h	H4	Block height = 4 rows													
	03h	H8	Block height = 8 rows														
19:2	Reserved																
	Format: Ignore Ignored																
1:0	Block Width																
	Format: Enumeration Width in Dwords of block being accessed. Range = [0,3] representing 1 to 8 Dwords.																
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>W1</td> <td>Block width = 1 Dword</td> </tr> <tr> <td>1h</td> <td>W2</td> <td>Block width = 2 Dwords</td> </tr> <tr> <td>2h</td> <td>W4</td> <td>Block width = 4 Dwords</td> </tr> <tr> <td>03h</td> <td>W8</td> <td>Block width = 8 Dwords</td> </tr> </tbody> </table>	Value	Name	Description	0h	W1	Block width = 1 Dword	1h	W2	Block width = 2 Dwords	2h	W4	Block width = 4 Dwords	03h	W8	Block width = 8 Dwords	
	Value	Name	Description														
	0h	W1	Block width = 1 Dword														
	1h	W2	Block width = 2 Dwords														
2h	W4	Block width = 4 Dwords															
03h	W8	Block width = 8 Dwords															

Block Message Header

MH_BTS_GO - Block Message Header		
Source:	DataPort 0	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0-1	63:0	Reserved
		Format: Ignore Ignored
2	31:0	Global Offset
		Format: U32
		Specifies the global element index into the buffer, in units of Hwords, Owords, Dwords, or Bytes (depending on the message).
		Programming Notes
		The Global Offset for the Aligned Block operations is specified as a Dword-aligned byte offset (offset bits [1:0] = 0), Oword-aligned byte offset (offset bits [3:0]=0), or Hword-aligned byte offset (offset bits [4:0]=0).
		If the address offset calculated with the Global Offset is greater than the Surface Size, then the access is Out-of-Bounds.
3-7	159:0	Reserved
		Format: Ignore Ignored

BR00 - BLT Opcode and Control

BR00 - BLT Opcode and Control									
Source:	BlitterCS								
Size (in bits):	32								
Default Value:	0x00000000								
DWord	Bit	Description							
0	31	BLT Engine Busy This bit indicates whether the BLT Engine is busy (1) or idle (0). This bit is replicated in the SETUP BLT Opcode and Control register. <table border="1" data-bbox="334 648 1471 787"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Idle [Default]</td> </tr> <tr> <td>1</td> <td>Busy</td> </tr> </tbody> </table>	Value	Name	0	Idle [Default]	1	Busy	
		Value	Name						
		0	Idle [Default]						
		1	Busy						
	30	Setup Instruction Instruction <table border="1" data-bbox="334 831 1471 877"> <tr> <td>Default Value:</td> <td>0</td> </tr> </table> The current instruction performs clipping (1).	Default Value:	0					
	Default Value:	0							
	29	Setup Monochrome Pattern This bit is decoded from the Setup instruction opcode to identify whether a color (0) or monochrome (1) pattern is used with the SCANLINE_BLT instruction. <table border="1" data-bbox="334 1031 1471 1169"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Color [Default]</td> </tr> <tr> <td>1</td> <td>Monochrome</td> </tr> </tbody> </table>	Value	Name	0	Color [Default]	1	Monochrome	
Value	Name								
0	Color [Default]								
1	Monochrome								
28:22	Instruction Target (Opcode) <table border="1" data-bbox="334 1213 1471 1260"> <tr> <td>Default Value:</td> <td>0000000b</td> </tr> </table> This is the contents of the Instruction Target field from the last BLT instruction. This field is used by the BLT Engine state machine to identify the BLT instruction it is to perform. The opcode specifies whether the source and pattern operands are color or monochrome.	Default Value:	0000000b						
Default Value:	0000000b								
21:20	32bpp Byte Mask This field is only used for 32bpp. <table border="1" data-bbox="334 1446 1471 1627"> <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	Monochrome Source Start <table border="1" data-bbox="334 1677 1471 1724"> <tr> <td>Default Value:</td> <td>000b</td> </tr> </table> This field indicates the starting monochrome pixel bit position within a byte per scan line of the source operand. The monochrome source is word aligned which means that at the end of the scan line all bits should be discarded until the next word boundary.	Default Value:	000b						
	Default Value:	000b							

BR00 - BLT Opcode and Control		
16	Bit/Byte Packed	
	Byte packed is for the NT driver.	
	Value	Name
	0b	Bit [Default]
	1b	Byte
15	Src Tiling Enable	
	Value	Name
	0b	Tiling Disabled (Linear) [Default]
	1b	Tiling enabled: Tile-X or Tile-Y
14:12	Horizontal Pattern Seed	
	Default Value:	0b
	This field indicates the pattern pixel position which corresponds to X = 0.	
11	Dest Tiling Enable	
	When set to '1', this means that Blitter is executing in Tiled mode. If '0' it means that Blitter is in Linear mode. Pre-Dev Blitter never executes in Tiled-Y mode, DevGT+ Blitter supports both Tile-X and Tile-Y modes. On reset, this bit will be '0'. This definition applies to only X, Y Blits.	
	Value	Name
	0b	Tiling Disabled (Linear blit) [Default]
	1b	Tiling enabled: Tile-X or Tile-Y
10:8	Transparency Range Mode	
	These bits control whether or not the byte(s) at the destination corresponding to a given pixel will be conditionally written, and what those conditions are. This feature can make it possible to perform various masking functions in order to selectively write or preserve graphics data already at the destination.	
	Value	Name
	xx0b	[Default]
	001b	Description
		No color transparency mode enabled. This causes normal operation with regard to writing data to the destination.
		[Source color transparency] The Transparency Color Low: (Pixel Greater or Equal) (source background register) and the Transparency Color High: (Pixel Less or Equal) (source foreground register) are compared to the source pixels. The range comparisons are done on each component (R, G, B) and then logically ANDed. If the source pixel components are not within the range defined by the Transparency Color registers, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation.

BR00 - BLT Opcode and Control		
	011b	[Source and Alpha color transparency] The Transparency Color Low: (Pixel Greater or Equal) (source background register) and the Transparency Color High: (Pixel Less or Equal) (source foreground register) are compared to the source pixels. The range comparisons are done on each component (A, R, G, B) and then logically ANDed. If the source pixel components are not within the range defined by the Transparency Color registers, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation."
	101b	[Destination and Alpha color transparency] The Transparency Color Low: (Pixel Greater or Equal) (source background register) and the Transparency Color High: (Pixel Less or Equal) (source foreground register) are compared to the destination pixels. The range comparisons are done on each component (A, R, G, B) and then logically ANDed. If the destination pixels are within the range, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation.
	111b	[Destination color transparency] The Transparency Color Low: (Pixel Greater or Equal) (source background register) and the Transparency Color High: (Pixel Less or Equal) (source foreground register) are compared to the destination pixels. The range comparisons are done on each component (R, G, B) and then logically ANDed. If the destination pixels are within the range, then the byte(s) at the destination corresponding to the current pixel are written with the result of the bit-wise operation.
7:5	Pattern Vertical Seed	
	Default Value:	000b
	This field specifies the pattern scan line which corresponds to Y=0.	
4	Destination Read Modify Write	
	Default Value:	0b
	This bit is decoded from the last instruction's opcode field and Destination Transparency Mode to identify whether a Destination read is needed.	
3	Color Source	
	Default Value:	0b
	This bit is decoded from the last instructions opcode field to identify whether a color (1) source is used.	
2	Monochrome Source	
	Default Value:	0b
	This bit is decoded from the last instructions opcode field to identify whether a monochrome (1) source is used.	
1	Color Pattern	
	Default Value:	0b
	This bit is decoded from the last instructions opcode field to identify whether a color (1) pattern is used.	

BR00 - BLT Opcode and Control			
0	<p>Monochrome Pattern</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Default Value:</td> <td style="width: 20%; text-align: center;">0b</td> </tr> </table> <p>This bit is decoded from the last instructions opcode field to identify whether a monochrome (1) pattern is used.</p>	Default Value:	0b
Default Value:	0b		

BR01 - Setup BLT Raster OP, Control, and Destination Offset

BR01 - Setup BLT Raster OP, Control, and Destination Offset											
Source:	BlitterCS										
Size (in bits):	32										
Default Value:	0x00000000										
DWord	Bit	Description									
0	31	<p>Solid Pattern Select</p> <p>This bit applies only when the pattern data is monochrome. This bit determines whether or not the BLT Engine actually performs read operations from the frame buffer in order to load the pattern data. Use of this feature to prevent these read operations can increase BLT Engine performance, if use of the pattern data is indeed not necessary. The BLT Engine is configured to accept either monochrome or color pattern data via the opcode field.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>[Default]</td> <td>This causes normal operation with regard to the use of the pattern data. The BLT Engine proceeds with the process of reading the pattern data, and the pattern data is used as the pattern operand for all bit-wise operations.</td> </tr> <tr> <td>1b</td> <td></td> <td>The BLT Engine forgoes the process of reading the pattern data, the presumption is made that all of the bits of the pattern data are set to 0, and the pattern operand for all bit-wise operations is forced to the background color specified in the Color Expansion Background Color Register.</td> </tr> </tbody> </table>	Value	Name	Description	0b	[Default]	This causes normal operation with regard to the use of the pattern data. The BLT Engine proceeds with the process of reading the pattern data, and the pattern data is used as the pattern operand for all bit-wise operations.	1b		The BLT Engine forgoes the process of reading the pattern data, the presumption is made that all of the bits of the pattern data are set to 0, and the pattern operand for all bit-wise operations is forced to the background color specified in the Color Expansion Background Color Register.
		Value	Name	Description							
		0b	[Default]	This causes normal operation with regard to the use of the pattern data. The BLT Engine proceeds with the process of reading the pattern data, and the pattern data is used as the pattern operand for all bit-wise operations.							
1b		The BLT Engine forgoes the process of reading the pattern data, the presumption is made that all of the bits of the pattern data are set to 0, and the pattern operand for all bit-wise operations is forced to the background color specified in the Color Expansion Background Color Register.									
30		<p>Clipping Enabled</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>[Default]</td> </tr> <tr> <td>1b</td> <td></td> </tr> </tbody> </table>	Value	Name	0b	[Default]	1b				
		Value	Name								
		0b	[Default]								
1b											
29		<p>Monochrome Source Transparency Mode</p> <p>This bit applies only when the source data is in monochrome. This bit determines whether or not the byte(s) at the destination corresponding to the pixel to which a given bit of the source data also corresponds will actually be written if that source data bit has the value of 0. This feature can make it possible to use the source as a transparency mask. The BLT Engine is configured to accepted either monochrome or color source data via the opcode field.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>[Default]</td> <td>This causes normal operation with regard to the use of the source data. Wherever a bit in the source data has the value of 0, the color specified in the background color register is used as the source operand in the bit-wise operation for the pixel corresponding to the source data bit, and the bytes at the destination corresponding to that pixel are written with the result.</td> </tr> <tr> <td>1b</td> <td></td> <td>Wherever a bit in the source data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the source data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.</td> </tr> </tbody> </table>	Value	Name	Description	0b	[Default]	This causes normal operation with regard to the use of the source data. Wherever a bit in the source data has the value of 0, the color specified in the background color register is used as the source operand in the bit-wise operation for the pixel corresponding to the source data bit, and the bytes at the destination corresponding to that pixel are written with the result.	1b		Wherever a bit in the source data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the source data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.
		Value	Name	Description							
		0b	[Default]	This causes normal operation with regard to the use of the source data. Wherever a bit in the source data has the value of 0, the color specified in the background color register is used as the source operand in the bit-wise operation for the pixel corresponding to the source data bit, and the bytes at the destination corresponding to that pixel are written with the result.							
1b		Wherever a bit in the source data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the source data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.									

BR01 - Setup BLT Raster OP, Control, and Destination Offset

28	Monochrome Pattern Transparency Mode	<p>This bit applies only when the pattern data is monochrome. This bit determines whether or not the byte(s) at the destination corresponding to the pixel to which a given bit of the pattern data also corresponds will actually be written if that pattern data bit has the value of 1. This feature can make it possible to use the pattern as a transparency mask. The BLT Engine is configured to accepted either monochrome or color pattern data via the opcode field.</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;">0b</td> <td style="text-align: center;">[Default]</td> <td>This causes normal operation with regard to the use of the pattern data. Wherever a bit in the pattern data has the value of 0, the color specified in the background color register is used as the pattern operand in the bit-wise operation for the pixel corresponding to the pattern data bit, and the bytes at the destination corresponding to that pixel are written with the result.</td> </tr> <tr> <td style="text-align: center;">1b</td> <td></td> <td>Wherever a bit in the pattern data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the pattern data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.</td> </tr> </tbody> </table>	Value	Name	Description	0b	[Default]	This causes normal operation with regard to the use of the pattern data. Wherever a bit in the pattern data has the value of 0, the color specified in the background color register is used as the pattern operand in the bit-wise operation for the pixel corresponding to the pattern data bit, and the bytes at the destination corresponding to that pixel are written with the result.	1b		Wherever a bit in the pattern data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the pattern data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.	
Value	Name	Description										
0b	[Default]	This causes normal operation with regard to the use of the pattern data. Wherever a bit in the pattern data has the value of 0, the color specified in the background color register is used as the pattern operand in the bit-wise operation for the pixel corresponding to the pattern data bit, and the bytes at the destination corresponding to that pixel are written with the result.										
1b		Wherever a bit in the pattern data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the pattern data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.										
27:26	32bpp Byte Mask	<p>This bit applies only when the pattern data is monochrome. This bit determines whether or not the byte(s) at the destination corresponding to the pixel to which a given bit of the pattern data also corresponds will actually be written if that pattern data bit has the value of 1. This feature can make it possible to use the pattern as a transparency mask. The BLT Engine is configured to accepted either monochrome or color pattern data via the opcode field.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">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											
25:24	Color Depth	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>8 Bit Color Depth [Default]</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>16 Bit Color Depth</td> </tr> <tr> <td style="text-align: center;">10b</td> <td>Alternate 16 Bit Color Depth</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>32 Bit Color Depth</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color Depth [Default]	01b	16 Bit Color Depth	10b	Alternate 16 Bit Color Depth	11b	32 Bit Color Depth
Value	Name											
00b	8 Bit Color Depth [Default]											
01b	16 Bit Color Depth											
10b	Alternate 16 Bit Color Depth											
11b	32 Bit Color Depth											
23:16	Raster Operation Select	<p>These 8 bits are used to select which one of 256 possible raster operations is to be performed by the BLT Engine.</p>										

BR01 - Setup BLT Raster OP, Control, and Destination Offset

	15:0	<p>Destination Pitch (Offset)</p> <p>For non-XY Blits, the signed 16bit field allows for specifying upto + 32Kbytes signed pitches in bytes (same as before). For X, Y Blits with tiled-X surfaces, the pitch for Destination will be 512Byte aligned and should be programmable upto + 128Kbytes. For X, Y Blits with tiled-Y surfaces, the pitch for Destination will be 128Byte aligned and should be programmable upto + 128Kbytes. In this case, this 16bit signed pitch field is used to specify upto + 32KDWords. For X, Y blits with nontiled surfaces (linear surfaces), this 16bit field can be programmed to byte specification of upto + 32Kbytes (same as before). These 16 bits store the signed memory address offset value by which the destination address originally specified in the Destination Address Register is incremented or decremented as each scan line's worth of destination data is written into the frame buffer by the BLT Engine, so that the destination address will point to the next memory address to which the next scan line's worth of destination data is to be written. If the intended destination of a BLT operation is within on-screen frame buffer memory, this offset is normally set so that each subsequent scan line's worth of destination data lines up vertically with the destination data in the scan line, above. However, if the intended destination of a BLT operation is within off-screen memory, this offset can be set so that each subsequent scan line's worth of destination data is stored at a location immediately after the location where the destination data for the last scan line ended, in order to create a single contiguous block of bytes of destination data at the destination.</p>
--	------	--

BR05 - Setup Expansion Background Color

BR05 - Setup Expansion Background Color		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:0	Setup Expansion Background Color Bits These bits provide the one, two, or four bytes worth of color data that select the background color to be used in the color expansion of monochrome pattern or source data for either the SCANLINE_BLT or TEXT_BLT instructions. BR05 is also used as the solid pattern for the PIXEL_BLT instruction. Whether one, two, or three bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.

BR06 - Setup Expansion Foreground Color

BR06 - Setup Expansion Foreground Color		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:0	<p>Setup Expansion Foreground Color Bits</p> <p>These bits provide the one, two, or four bytes worth of color data that select the foreground color to be used in the color expansion of monochrome pattern or source data for either the SCANLINE_BLT or TEXT_BLT instructions. Whether one, two, or three bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.</p>

BR07 - Setup Blit Color Pattern Address Lower Order Address bits

BR07 - Setup Blit Color Pattern Address Lower Order Address bits				
Source:	BlitterCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:6	<p>Setup Blit Color Pattern Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Lower 32bits of the 48bit addressing. These 26 bits specify the starting address of the (8X8) pixel color pattern from the SETUP_BLT instruction. This register works identically to the Pattern Address register (BR15), but this version is only used with the SCANLINE_BLT instruction execution (the actual programming for this, is done in XY_SETUP_BLT command). The pattern data must be located in linear memory. The pattern data must be located on a pattern-size boundary. The pattern is always of 8x8 pixels, and therefore, its size is dependent upon its pixel depth. The pixel depth may be 8, 16, or 32 bits per pixel if the pattern is in color (the pixel depth of a color pattern must match the pixel depth to which the graphics system has been set). Monochrome patterns require 8 bytes and is supplied through the instruction. Color patterns of 8, 16, and 32 bits per pixel color depth must start on 64-byte, 128-byte and 256-byte boundaries, respectively. The Pattern Base Address programmed, must always be Cache Line (64byte) aligned.</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			

BR09 - Destination Address Lower Order Address Bits

BR09 - Destination Address Lower Order Address Bits				
Source:	BlitterCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:0	<p>Destination Address Bits</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:0]</td> </tr> </table> <p>When tiling is enabled for XY-blits, this base address should be limited to 4KB. when tiling is disabled for XY-blits, this base address should be CL (64byte) aligned. These lower 32bits of the 48bit address, which specify the starting pixel address of the destination data. This register is also the working destination address register for the lower 32bits of the address, and changes as the BLT Engine performs the accesses. Used as the scan line address (Destination Y Address and Destination Y1 Address) for BLT instructions: PIXEL_BLT, SCANLINE_BLT, and TEXT_BLT. In this case the address points to the first pixel in a scan line and is compared with the ClipRect Y1 and Y2 address registers to determine whether the scan line should be written or not. The Destination Y1 address is the top scan line to be written for text. Note that for non-XY blits (COLOR_BLT, SRC_COPY_BLT), this address points to the first byte to be written. Note: Some instructions affect only one scan line (requiring only one coordinate); other instructions affect multiple scan lines and need both coordinates.</p>	Format:	GraphicsAddress[31:0]
Format:	GraphicsAddress[31:0]			

BR11 - BLT Source Pitch (Offset)

BR11 - BLT Source Pitch (Offset)		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:16	Reserved
	15:0	<p>Source Pitch (Offset)</p> <p>For non-XY Blits with color source operand (SRC_COPY_BLT), the signed 16bit field allows for specifying upto + 32Kbytes signed pitch in bytes (same as before). For X, Y Blits with tiled-X surfaces, the pitch for Color Source will be 512Byte aligned and should be programmable upto + 128Kbytes. For X, Y Blits with tiled-Y surfaces, the pitch for Color Source will be 128Byte aligned and should be programmable upto + 128Kbytes. In this case, this 16bit signed pitch field is used to specify upto + 32KDWords. For X, Y blits with nontiled color source surfaces (linear surfaces), this 16bit field can be programmed to byte specification of upto + 32Kbytes (same as before). When the color source data is located within the frame buffer or AGP aperture, these signed 16 bits store the memory address offset (pitch) value by which the source address originally specified in the Source Address Register is incremented or decremented as each scan line's worth of source data is read from the frame buffer by the BLT Engine, so that the source address will point to the next memory address from which the next scan line's worth of source data is to be read. Note that if the intended source of a BLT operation is within on-screen frame buffer memory, this offset is normally set to accommodate the fact that each subsequent scan line's worth of source data lines up vertically with the source data in the scan line, above. However, if the intended source of a BLT operation is within off-screen memory, this offset can be set to accommodate a situation in which the source data exists as a single contiguous block of bytes where in each subsequent scan line's worth of source data is stored at a location immediately after the location where the source data for the last scan line ended.</p>

BR12 - Source Address Lower order Address bits

BR12 - Source Address Lower order Address bits				
Source:	BlitterCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:0	<p>Source Address Bits</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:0]</td> </tr> </table> <p>Lower 32bits of the 48bit addressing. When tiling is enabled for XY-blits with Color source surfaces, this base address should be limited to 4KB. When tiling is disabled for XY-blits, this base address should be CL (64byte) aligned. Note that for non-XY blit with Color Source (SRC_COPY_BLT), this address points to the first byte to be read. These lower 32bits of the 48bit address, specify the starting pixel address of the color source data. The lower 3 bits are used to indicate the position of the first valid byte within the first Quadword of the source data. If this Source happens to be a Monosource surface, then this Monosource Base Address programmed, must always be Cache Line (64byte) aligned.</p>	Format:	GraphicsAddress[31:0]
Format:	GraphicsAddress[31:0]			

BR13 - BLT Raster OP, Control, and Destination Pitch

BR13 - BLT Raster OP, Control, and Destination Pitch											
Source:	BlitterCS										
Size (in bits):	32										
Default Value:	0x00000000										
DWord	Bit	Description									
0	31	<p>Solid Pattern Select</p> <p>This bit applies only when the pattern data is monochrome. This bit determines whether or not the BLT Engine actually performs read operations from the frame buffer in order to load the pattern data. Use of this feature to prevent these read operations can increase BLT Engine performance, if use of the pattern data is indeed not necessary. The BLT Engine is configured to accept either monochrome or color pattern data via the opcode field.</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>This causes normal operation with regard to the use of the pattern data. The BLT Engine proceeds with the process of reading the pattern data, and the pattern data is used as the pattern operand for all bit-wise operations.</td> </tr> <tr> <td>1</td> <td></td> <td>The BLT Engine forgoes the process of reading the pattern data, the presumption is made that all of the bits of the pattern data are set to 0, and the pattern operand for all bit-wise operations is forced to the background color specified in the Color Expansion Background Color Register.</td> </tr> </tbody> </table>	Value	Name	Description	0	[Default]	This causes normal operation with regard to the use of the pattern data. The BLT Engine proceeds with the process of reading the pattern data, and the pattern data is used as the pattern operand for all bit-wise operations.	1		The BLT Engine forgoes the process of reading the pattern data, the presumption is made that all of the bits of the pattern data are set to 0, and the pattern operand for all bit-wise operations is forced to the background color specified in the Color Expansion Background Color Register.
		Value	Name	Description							
		0	[Default]	This causes normal operation with regard to the use of the pattern data. The BLT Engine proceeds with the process of reading the pattern data, and the pattern data is used as the pattern operand for all bit-wise operations.							
1		The BLT Engine forgoes the process of reading the pattern data, the presumption is made that all of the bits of the pattern data are set to 0, and the pattern operand for all bit-wise operations is forced to the background color specified in the Color Expansion Background Color Register.									
30	<p>Clipping Enabled</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> </table>	Default Value:	0								
Default Value:	0										
29		<p>Monochrome Source Transparency Mode</p> <p>This bit applies only when the source data is in monochrome. This bit determines whether or not the byte(s) at the destination corresponding to the pixel to which a given bit of the source data also corresponds will actually be written if that source data bit has the value of 0. This feature can make it possible to use the source as a transparency mask. The BLT Engine is configured to accepted either monochrome or color source data via the opcode field.</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>This causes normal operation with regard to the use of the source data. Wherever a bit in the source data has the value of 0, the color specified in the background color register is used as the source operand in the bit-wise operation for the pixel corresponding to the source data bit, and the bytes at the destination corresponding to that pixel are written with the result.</td> </tr> <tr> <td>1</td> <td></td> <td>Where a bit in the source data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the source data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.</td> </tr> </tbody> </table>	Value	Name	Description	0	[Default]	This causes normal operation with regard to the use of the source data. Wherever a bit in the source data has the value of 0, the color specified in the background color register is used as the source operand in the bit-wise operation for the pixel corresponding to the source data bit, and the bytes at the destination corresponding to that pixel are written with the result.	1		Where a bit in the source data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the source data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.
		Value	Name	Description							
		0	[Default]	This causes normal operation with regard to the use of the source data. Wherever a bit in the source data has the value of 0, the color specified in the background color register is used as the source operand in the bit-wise operation for the pixel corresponding to the source data bit, and the bytes at the destination corresponding to that pixel are written with the result.							
1		Where a bit in the source data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the source data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.									

BR13 - BLT Raster OP, Control, and Destination Pitch

28	<p>Monochrome Pattern Transparency Mode</p> <p>This bit applies only when the pattern data is monochrome. This bit determines whether or not the byte(s) at the destination corresponding to the pixel to which a given bit of the pattern data also corresponds will actually be written if that pattern data bit has the value of 1. This feature can make it possible to use the pattern as a transparency mask. The BLT Engine is configured to accepted either monochrome or color pattern data via the opcode in the Opcode and Control register.</p>	
	Value	Name
	0	[Default]
	1	

Value	Name	Description
0	[Default]	This causes normal operation with regard to the use of the pattern data. Where a bit in the pattern data has the value of 0, the color specified in the background color register is used as the pattern operand in the bit-wise operation for the pixel corresponding to the pattern data bit, and the bytes at the destination corresponding to that pixel are written with the result.
1		Wherever a bit in the pattern data has the value of 0, the byte(s) at the destination corresponding to the pixel to which the pattern data bit also corresponds are simply not written, and the data at those byte(s) at the destination are allowed to remain unchanged.

27:26	<p>32bpp Byte Mask</p> <p>This field is only used for 32bpp.</p>	
	Value	Name
	00b	[Default]
	1xb	Write Alpha Channel
	x1b	Write RGB Channel

25:24	<p>Color Depth</p>	
	Value	Name
	00b	8 Bit Color Depth [Default]
	01b	16 Bit Color Depth
	10b	24 Bit Color Depth
	11b	Reserved

23:16	<p>Raster Operation Select</p>	
	Default Value:	00000000b
	<p>These 8 bits are used to select which one of 256 possible raster operations is to be performed by the BLT Engine.</p>	

BR13 - BLT Raster OP, Control, and Destination Pitch

	15:0	<p>Destination Pitch(Offset)</p> <p>These 16 bits store the signed memory address offset value by which the destination address originally specified in the Destination Address Register is incremented or decremented as each scan line's worth of destination data is written into the frame buffer by the BLT Engine, so that the destination address will point to the next memory address to which the next scan line's worth of destination data is to be written. If the intended destination of a BLT operation is within on-screen frame buffer memory, this offset is normally set so that each subsequent scan line's worth of destination data lines up vertically with the destination data in the scan line, above. However, if the intended destination of a BLT operation is within off-screen memory, this offset can be set so that each subsequent scan line's worth of destination data is stored at a location immediately after the location where the destination data for the last scan line ended, in order to create a single contiguous block of bytes of destination data at the destination.</p>
--	------	--

BR14 - Destination Width and Height

BR14 - Destination Width and Height		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
<p>BR14 contains the values for the height and width of the data to be BLT. If these values are not correct, such that the BLT Engine is either expecting data it does not receive or receives data it did not expect, the system can hang.</p>		
DWord	Bit	Description
0	31:29	Reserved
	28:16	Destination Height These 13 bits specify the height of the destination data in terms of the number of scan lines. This is a working register.
	15:13	Reserved
	12:0	Destination Byte Width These 13 bits specify the width of the destination data in terms of the number of bytes per scan line. The number of pixels per scan line into which this value translates depends upon the color depth to which the graphics system has been set.

BR15 - Color Pattern Address Lower order Address bits

BR15 - Color Pattern Address Lower order Address bits				
Source:	BlitterCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:6	<p>Color Pattern Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Lower 32bits of the 48bit addressing. There is no change to the Color Pattern address specification due to Non-Power-of-2 change. It remains the same as before. The pattern data must be located in linear memory. These 26 bits specify the starting address of the (8X8) pixel color pattern. The pattern data must be located on a pattern-size boundary. The pattern is always of 8x8 pixels, and therefore, its size is dependent upon its pixel depth. The pixel depth may be 8, 16, or 32 bits per pixel if the pattern is in color (the pixel depth of a color pattern must match the pixel depth to which the graphics system has been set). Monochrome patterns require 8 bytes and are applied through the instruction. Color patterns of 8, 16, and 32 bits per pixel color depth must start on 64-byte, 128-byte and 256-byte boundaries, respectively. The Pattern Base Address programmed, must always be Cache Line (64byte) aligned.</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			

BR16 - Pattern Expansion Background and Solid Pattern Color

BR16 - Pattern Expansion Background and Solid Pattern Color		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:0	<p>Pattern Expansion Background Color Bits</p> <p>These bits provide the one, two, or four bytes worth of color data that select the background color to be used in the color expansion of monochrome pattern data during BLT operations. Whether one, two, or four bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.</p>

BR17 - Pattern Expansion Foreground Color

BR17 - Pattern Expansion Foreground Color		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:0	Pattern Expansion Background Color Bits These bits provide the one, two, or four bytes worth of color data that select the foreground color to be used in the color expansion of monochrome pattern data during BLT operations. Whether one, two, or four bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.

BR18 - Source Expansion Background and Destination Color

BR18 - Source Expansion Background and Destination Color		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:0	<p>Source Expansion Background Color Bits</p> <p>These bits provide the one, two, or four bytes worth of color data that select the background color to be used in the color expansion of monochrome source data during BLT operations. This register is also used to support destination transparency mode and Solid color fill. Whether one, two, three, or four bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.</p>

BR19 - Source Expansion Foreground Color

BR19 - Source Expansion Foreground Color		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:0	Pattern/Source Expansion Foreground Color Bits These bits provide the one, two, or four bytes worth of color data that select the foreground color to be used in the color expansion of monochrome source data during BLT operations. Whether one, two, or four bytes worth of color data is needed depends upon the color depth to which the BLT Engine has been set. For a color depth of 32bpp, 16bpp and 8bpp, bits [31:0], [15:0] and [7:0], respectively, are used.

BR27 - Destination Address Higher Order Address

BR27 - Destination Address Higher Order Address				
Source:	BlitterCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
15:0	<p>Destination Address Upper DWORD</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>When tiling is enabled for XY-blits, this base address should be limited to 4KB. Otherwise for XY blits, there is no restriction and it is same as before. These upper 16bits of the 48bit address, along with BR09 register, will specify the starting pixel address of the destination data. This register is also the working destination address register for the upper 16bits of the destination address, and changes as the BLT Engine performs the accesses. Used as the scan line address (Destination Y Address and Destination Y1 Address) for BLT instructions: PIXEL_BLT, SCANLINE_BLT, and TEXT_BLT. In this case the address points to the first pixel in a scan line and is compared with the ClipRect Y1 and Y2 address registers to determine whether the scan line should be written or not. The Destination Y1 address is the top scan line to be written for text. Note that for non-XY blits (COLOR_BLT, SRC_COPY_BLT), this 16bits of the 48bit address, along with BR09 register, points to the first byte to be written. This register is always the last register written for a BLT drawing instruction. Writing BR27 starts the BLT engine execution. Note: Some instructions affect only one scan line (requiring only one coordinate); other instructions affect multiple scan lines and need both coordinates.</p>	Format:	GraphicsAddress[47:32]	
Format:	GraphicsAddress[47:32]			

BR28 - Source Address Higher order Address

BR28 - Source Address Higher order Address				
Source:	BlitterCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:16	Reserved <table border="1" data-bbox="332 611 1471 657"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
15:0	Source Address Upper DWORD <table border="1" data-bbox="332 703 1471 749"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>These upper 16bits of the 48bit address, specify the starting pixel address of the color or mono source data. When tiling is enabled for XY-blits with Color source surfaces, this base address should be limited to 4KB. Otherwise for XY blits, there is no restriction and it is same as before, including for monosource and text blits. Note that for non-XY blit with Color Source (SRC_COPY_BLT), this address points to the first byte to be read.</p>	Format:	GraphicsAddress[47:32]	
Format:	GraphicsAddress[47:32]			

BR29 - Color Pattern Address Higher order Address

BR29 - Color Pattern Address Higher order Address				
Source:	BlitterCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:16	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
15:0	Color Pattern Address Upper DWORD Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>GraphicsAddress[47:32]</td></tr></table> These upper 16bits of the 48bit address,specify the starting address of the (8X8) pixel pattern.		GraphicsAddress[47:32]	
	GraphicsAddress[47:32]			

BR30 - Setup Blit Color Pattern Address Higher Order Address

BR30 - Setup Blit Color Pattern Address Higher Order Address				
Source:	BlitterCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:16	Reserved <table border="1" data-bbox="337 611 1468 657"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
15:0	Setup Blit Color Pattern Address Upper DWORD <table border="1" data-bbox="337 703 1468 749"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>These upper 16bits of the 48bit address,specify the starting address of the (8X8) pixel pattern.</p>	Format:	GraphicsAddress[47:32]	
Format:	GraphicsAddress[47:32]			

Byte Masked Media Block Message Header

MH_MBBM - Byte Masked Media Block Message Header		
Source:	DataPort 1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	X Offset
		Format: S31
		X offset (in bytes) of the upper left corner of the block into the surface.
		Programming Notes
Must be DWord aligned (Bits 1:0 MBZ) for the write form of the message.		
1	31:0	Y Offset
		Format: S31 Y offset (in rows) of the upper left corner of the block into the surface.
2	31:0	Media Block Message Control
		Format: MHC_MBBM_CONTROL Specifies the Byte Masked message subtype and its additional input parameters.
3	31:0	Byte Mask
		Format: U32
		Specifies the Byte Mask for writes when Message Mode field is BYTE_MASK.
		Programming Notes
The Byte mask applies horizontally to each row of output: bit 0 for byte 0, through bit 31 for byte 31.		
4	31:0	FFTID
		Format: MHC_FFTID Fixed Function Thread ID
5-7	95:0	Reserved
		Format: Ignore Ignored

Byte Masked Media Block Message Header Control

MHC_MBBM_CONTROL - Byte Masked Media Block Message Header Control													
Source:	BSpec												
Size (in bits):	32												
Default Value:	0x00000000												
DWord	Bit	Description											
0	31:30	<p>Message Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> </table> <p>Specifies the Media Block Write Message subtype is Byte Masked.</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 style="text-align: center;">02h</td> <td style="text-align: center;">BYTE_MASK</td> <td>The Block Height and Block Width fields are specified in this Dword. The Byte Mask qualifies which bytes are written.</td> </tr> <tr> <td style="text-align: center;">Others</td> <td style="text-align: center;">Reserved</td> <td>Reserved.</td> </tr> </tbody> </table>	Format:	Enumeration	Value	Name	Description	02h	BYTE_MASK	The Block Height and Block Width fields are specified in this Dword. The Byte Mask qualifies which bytes are written.	Others	Reserved	Reserved.
	Format:	Enumeration											
	Value	Name	Description										
	02h	BYTE_MASK	The Block Height and Block Width fields are specified in this Dword. The Byte Mask qualifies which bytes are written.										
	Others	Reserved	Reserved.										
	29	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore									
	Format:	Ignore											
	28:24	<p>Sub-Register Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">U5</td> </tr> </table> <p>This field is ignored (reserved) for Media Block Write message.</p>	Format:	U5									
Format:	U5												
23:22	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore										
Format:	Ignore												
21:16	<p>Block Height</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">U6</td> </tr> </table> <p>Height in rows of block being accessed. Range = [0,63] representing 1 to 64 rows</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="text-align: center;">Restriction</th> </tr> </thead> <tbody> <tr> <td>If Block Width (bytes), then Maximum Block Height (rows) is constrained by (# Dwords width) * (# rows) <= 64 Dwords.</td> </tr> </tbody> </table>	Format:	U6	Restriction	If Block Width (bytes), then Maximum Block Height (rows) is constrained by (# Dwords width) * (# rows) <= 64 Dwords.								
Format:	U6												
Restriction													
If Block Width (bytes), then Maximum Block Height (rows) is constrained by (# Dwords width) * (# rows) <= 64 Dwords.													
15:10	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore										
Format:	Ignore												
9:8	<p>Register Pitch Control</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">U2</td> </tr> </table> <p>This field is ignored (reserved) for a Media Block Write message.</p>	Format:	U2										
Format:	U2												

MHC_MBBM_CONTROL - Byte Masked Media Block Message Header Control

	7:6	Reserved	
		Format:	Ignore
		Ignored	
	5:0	Block Width	
		Format:	U6
		Width in bytes of the block being accessed. Range = [0,31] representing 1 to 32 Bytes.	
		Programming Notes	
		Must be DWord aligned for Media Block Write message.	

CC_VIEWPORT

CC_VIEWPORT					
Source:	BSpec				
Size (in bits):	64				
Default Value:	0x00000000, 0x00000000				
<p>The viewport state is stored as an array of up to 16 elements, each of which contains the DWords described here. The start of each element is spaced 2 DWords apart. The first element of the viewport state array is aligned to a 32-byte boundary. The Minimum Depth must be greater than or equal to zero on D16_UNORM, D24_UNORM_X8_UINT, or D24_UNORM_S8_UINT depth formats. The Minimum Depth must be greater than or equal to -1.0 for D32_FLOAT_S8X24_UINT or D32_FLOAT formats. The Maximum Depth must be less than or equal to +1.0. The max must be greater than or equal to the min.</p>					
DWord	Bit	Description			
0	31:0	<p>Minimum Depth</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>Indicates the minimum depth. The interpolated or computed depth is clamped to this value prior to the depth test.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>The Minimum depth value must be less-than-or-equal to the Maximum depth value. The Minimum depth value cannot be NAN (Not-A-Number). The Minimum depth value must not be less than -1.0.</p>	Format:	IEEE_Float	Programming Notes
Format:	IEEE_Float				
Programming Notes					
1	31:0	<p>Maximum Depth</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>Indicates the maximum depth. The interpolated or computed depth is clamped to this value prior to the depth test.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>The Maximum depth value cannot be NAN (Not-A-Number). The Maximum depth value must be less-than-or-equal to +1.0.</p>	Format:	IEEE_Float	Programming Notes
Format:	IEEE_Float				
Programming Notes					

Channel Mask Message Descriptor Control Field

MDC_CMASK - Channel Mask Message Descriptor Control Field																																																							
Source:	BSpec																																																						
Size (in bits):	4																																																						
Default Value:	0x00000000																																																						
DWord	Bit	Description																																																					
0	3:0	<p>Mask</p> <table border="1"> <tr> <td>Format:</td> <td>Enumeration</td> </tr> </table> <p>For the read message, indicates that which channels are read from the surface and included in the writeback message. For the write message, indicates which channels are included in the message payload and written to the surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>RGBA [Default]</td> <td>Red, Green, Blue, and Alpha are included</td> </tr> <tr> <td>01h</td> <td>GBA</td> <td>Green, Blue, and Alpha are included</td> </tr> <tr> <td>02h</td> <td>RBA</td> <td>Red, Blue, and Alpha are included</td> </tr> <tr> <td>03h</td> <td>BA</td> <td>Blue and Alpha are included</td> </tr> <tr> <td>04h</td> <td>RGA</td> <td>Red, Green, and Alpha are included</td> </tr> <tr> <td>05h</td> <td>GA</td> <td>Green and Alpha are included</td> </tr> <tr> <td>06h</td> <td>RA</td> <td>Red and Alpha are included</td> </tr> <tr> <td>07h</td> <td>A</td> <td>Alpha is included</td> </tr> <tr> <td>08h</td> <td>RGB</td> <td>Red, Green, and Blue are included</td> </tr> <tr> <td>09h</td> <td>GB</td> <td>Green and Blue are included</td> </tr> <tr> <td>0Ah</td> <td>RB</td> <td>Red and Blue are included</td> </tr> <tr> <td>0Bh</td> <td>B</td> <td>Blue is included</td> </tr> <tr> <td>0Ch</td> <td>RG</td> <td>Red and Green are included</td> </tr> <tr> <td>0Dh</td> <td>G</td> <td>Green is included</td> </tr> <tr> <td>0Eh</td> <td>R</td> <td>Red is included</td> </tr> <tr> <td>0Fh</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Format:	Enumeration	Value	Name	Description	00h	RGBA [Default]	Red, Green, Blue, and Alpha are included	01h	GBA	Green, Blue, and Alpha are included	02h	RBA	Red, Blue, and Alpha are included	03h	BA	Blue and Alpha are included	04h	RGA	Red, Green, and Alpha are included	05h	GA	Green and Alpha are included	06h	RA	Red and Alpha are included	07h	A	Alpha is included	08h	RGB	Red, Green, and Blue are included	09h	GB	Green and Blue are included	0Ah	RB	Red and Blue are included	0Bh	B	Blue is included	0Ch	RG	Red and Green are included	0Dh	G	Green is included	0Eh	R	Red is included	0Fh	Reserved	Ignored
Format:	Enumeration																																																						
Value	Name	Description																																																					
00h	RGBA [Default]	Red, Green, Blue, and Alpha are included																																																					
01h	GBA	Green, Blue, and Alpha are included																																																					
02h	RBA	Red, Blue, and Alpha are included																																																					
03h	BA	Blue and Alpha are included																																																					
04h	RGA	Red, Green, and Alpha are included																																																					
05h	GA	Green and Alpha are included																																																					
06h	RA	Red and Alpha are included																																																					
07h	A	Alpha is included																																																					
08h	RGB	Red, Green, and Blue are included																																																					
09h	GB	Green and Blue are included																																																					
0Ah	RB	Red and Blue are included																																																					
0Bh	B	Blue is included																																																					
0Ch	RG	Red and Green are included																																																					
0Dh	G	Green is included																																																					
0Eh	R	Red is included																																																					
0Fh	Reserved	Ignored																																																					

Channel Mode Message Descriptor Control Field

MDC_CMODE - Channel Mode Message Descriptor Control Field													
Source:	BSpec												
Size (in bits):	1												
Default Value:	0x00000000												
DWord	Bit	Description											
0	0	<p>Channel Mode</p> <table border="1"> <tr> <td>Format:</td> <td>Enumeration</td> </tr> </table> <p>Two modes of channel-enable are provided: a SIMD8 or SIMD16 Dword channel serial view of a register, and a SIMD4x2 view of a register.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Oword</td> <td>All 4 Dwords are read or written if one or more of these channels are enabled</td> </tr> <tr> <td>1</td> <td>Dword</td> <td>Each Dword is read or written only if its corresponding channel is enabled.</td> </tr> </tbody> </table>	Format:	Enumeration	Value	Name	Description	0	Oword	All 4 Dwords are read or written if one or more of these channels are enabled	1	Dword	Each Dword is read or written only if its corresponding channel is enabled.
Format:	Enumeration												
Value	Name	Description											
0	Oword	All 4 Dwords are read or written if one or more of these channels are enabled											
1	Dword	Each Dword is read or written only if its corresponding channel is enabled.											

Clock Gating Disable Format

Clock Gating Disable Format											
Source:	BSpec										
Size (in bits):	1										
Default Value:	0x00000000										
DWord	Bit	Description									
0	0	Clock_Gate_Disable <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Enable</td> <td>Clock gating controlled by unit logic</td> </tr> <tr> <td>1b</td> <td>Disable</td> <td>Disable clock gating function</td> </tr> </tbody> </table>	Value	Name	Description	0b	Enable	Clock gating controlled by unit logic	1b	Disable	Disable clock gating function
Value	Name	Description									
0b	Enable	Clock gating controlled by unit logic									
1b	Disable	Disable clock gating function									



Clock Gating Disable Format

Clock Gating Disable Format											
Source:	BSpec										
Size (in bits):	1										
Default Value:	0x00000000										
DWord	Bit	Description									
0	0	Clock Gate Disable <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 style="text-align: center;">Enable</td> <td>Clock gating controlled by unit enabling logic</td> </tr> <tr> <td style="text-align: center;">1b</td> <td style="text-align: center;">Disable</td> <td>Disable clock gating function</td> </tr> </tbody> </table>	Value	Name	Description	0b	Enable	Clock gating controlled by unit enabling logic	1b	Disable	Disable clock gating function
Value	Name	Description									
0b	Enable	Clock gating controlled by unit enabling logic									
1b	Disable	Disable clock gating function									

COLOR_CALC_STATE

COLOR_CALC_STATE										
Source:	BSpec									
Size (in bits):	192									
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000									
This definition applies to SKL devices. It is pointed to by a field in 3DSTATE_CC_STATE_POINTERS, and stored at a 64-byte aligned boundary.										
DWord	Bit	Description								
0	31:16	Reserved								
		Format: MBZ								
	15	Round Disable Function Disable								
		Format: Disable								
		Disables the round-disable function of the color calculator.								
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Cancelled</td> <td>Dithering is cancelled based on the data used by blend to avoid drift.</td> </tr> <tr> <td>1</td> <td>Not Cancelled</td> <td>Dithering is NOT cancelled.</td> </tr> </tbody> </table>		Value	Name	Description	0	Cancelled	Dithering is cancelled based on the data used by blend to avoid drift.	1	Not Cancelled	Dithering is NOT cancelled.
Value	Name	Description								
0	Cancelled	Dithering is cancelled based on the data used by blend to avoid drift.								
1	Not Cancelled	Dithering is NOT cancelled.								
14:1	Reserved									
	Format: MBZ									
0	Alpha Test Format									
	This field selects the format for Alpha Reference Value and the format in which Alpha Test is performed.									
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>ALPHATEST_UNORM8</td> <td>UNorm8</td> </tr> <tr> <td>1h</td> <td>ALPHATEST_FLOAT32</td> <td>Float32</td> </tr> </tbody> </table>	Value	Name	Description	0h	ALPHATEST_UNORM8	UNorm8	1h	ALPHATEST_FLOAT32	Float32
	Value	Name	Description							
	0h	ALPHATEST_UNORM8	UNorm8							
1h	ALPHATEST_FLOAT32	Float32								
Programming Notes										
Alpha-test format is independent of RT format. When PS outputs UNIT/SINT alpha-value, it will be treated as IEEE 32bit float number for the purpose of alpha-test.										
1	31:0	Alpha Reference Value As UNORM8								
		Exists If: [Alpha Test Format] == 'ALPHATEST_UNORM8'								
		Format: UNORM8 Upper 24 bits MBZ								
	This field specifies the alpha reference value to compare against in the Alpha Test function.									
	31:0	Alpha Reference Value As FLOAT32								
Exists If: [Alpha Test Format] == 'ALPHATEST_FLOAT32'										
Format: IEEE_Float										
This field specifies the alpha reference value to compare against in the Alpha Test function.										

COLOR_CALC_STATE		
2	31:0	Blend Constant Color Red
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>This field specifies the Red channel of the Constant Color used in Color Buffer Blending.</p>
Format:	IEEE_Float	
3	31:0	Blend Constant Color Green
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>This field specifies the Green channel of the Constant Color used in Color Buffer Blending.</p>
Format:	IEEE_Float	
4	31:0	Blend Constant Color Blue
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>This field specifies the Blue channel of the Constant Color used in Color Buffer Blending.</p>
Format:	IEEE_Float	
5	31:0	Blend Constant Color Alpha
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>This field specifies the Alpha channel of the Constant Color used in Color Buffer Blending.</p>
Format:	IEEE_Float	

COLOR_PROCESSING_STATE - ACE State

COLOR_PROCESSING_STATE - ACE State								
Source:	BSpec							
Size (in bits):	416							
Default Value:	0x00000068, 0x4C382410, 0x9C887460, 0xEBD8C4B0, 0x604C3824, 0xB09C8874, 0x0000D8C4, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000							
This state structure contains the ACE state used by the color processing function. It corresponds to DW29..DW41 of the Color Processing State.								
DWord	Bit	Description						
0	31:7	Reserved Format: MBZ						
	6:2	Skin Threshold Format: U5 Used for Y analysis (min/max) for pixels which are higher than skin threshold. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1-31</td> <td></td> </tr> <tr> <td>26</td> <td>[Default]</td> </tr> </tbody> </table>	Value	Name	1-31		26	[Default]
	Value	Name						
	1-31							
26	[Default]							
1	Full Image Histogram Default Value: 0 Format: Enable Used to ignore the area of interest for full image histogram.							
0	ACE Enable Format: Enable							
1	31:24	Y3 Default Value: 76 Format: U8 The value of the y_pixel for point 3 in PWL.						
	23:16	Y2 Default Value: 56 Format: U8 The value of the y_pixel for point 2 in PWL.						
	15:8	Y1 Default Value: 36 Format: U8 The value of the y_pixel for point 1 in PWL.						

COLOR_PROCESSING_STATE - ACE State						
	7:0	Ymin <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">16</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 0 in PWL.</p>	Default Value:	16	Format:	U8
		Default Value:	16			
Format:	U8					
2	31:24	Y7 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">156</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 7 in PWL.</p>	Default Value:	156	Format:	U8
		Default Value:	156			
	Format:	U8				
	23:16	Y6 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">136</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 6 in PWL.</p>	Default Value:	136	Format:	U8
		Default Value:	136			
	Format:	U8				
	15:8	Y5 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">116</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 5 in PWL.</p>	Default Value:	116	Format:	U8
		Default Value:	116			
Format:	U8					
7:0	Y4 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">96</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 4 in PWL.</p>	Default Value:	96	Format:	U8	
	Default Value:	96				
Format:	U8					
3	31:24	Ymax <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">235</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 11 in PWL.</p>	Default Value:	235	Format:	U8
		Default Value:	235			
	Format:	U8				
	23:16	Y10 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">216</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 10 in PWL.</p>	Default Value:	216	Format:	U8
Default Value:		216				
Format:	U8					
15:8	Y9 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">196</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 9 in PWL.</p>	Default Value:	196	Format:	U8	
	Default Value:	196				
Format:	U8					
7:0	Y8 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">176</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 8 in PWL.</p>	Default Value:	176	Format:	U8	
Default Value:	176					
Format:	U8					

COLOR_PROCESSING_STATE - ACE State						
4	31:24	B4 <table border="1"> <tr> <td>Default Value:</td> <td>96</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 4 in PWL.</p>	Default Value:	96	Format:	U8
		Default Value:	96			
		Format:	U8			
		23:16	B3 <table border="1"> <tr> <td>Default Value:</td> <td>76</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 3 in PWL.</p>	Default Value:	76	Format:
Default Value:	76					
Format:	U8					
15:8	B2 <table border="1"> <tr> <td>Default Value:</td> <td>56</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 2 in PWL.</p>	Default Value:	56	Format:	U8	
	Default Value:	56				
Format:	U8					
7:0	B1 <table border="1"> <tr> <td>Default Value:</td> <td>36</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 1 in PWL.</p>	Default Value:	36	Format:	U8	
Default Value:	36					
Format:	U8					
5	31:24	B8 <table border="1"> <tr> <td>Default Value:</td> <td>176</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 8 in PWL.</p>	Default Value:	176	Format:	U8
		Default Value:	176			
		Format:	U8			
		23:16	B7 <table border="1"> <tr> <td>Default Value:</td> <td>156</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 7 in PWL.</p>	Default Value:	156	Format:
Default Value:	156					
Format:	U8					
15:8	B6 <table border="1"> <tr> <td>Default Value:</td> <td>136</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 6 in PWL.</p>	Default Value:	136	Format:	U8	
	Default Value:	136				
Format:	U8					
7:0	B5 <table border="1"> <tr> <td>Default Value:</td> <td>116</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 5 in PWL.</p>	Default Value:	116	Format:	U8	
Default Value:	116					
Format:	U8					
6	31:16	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
		Format:	MBZ			
15:8	B10 <table border="1"> <tr> <td>Default Value:</td> <td>216</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 10 in PWL.</p>	Default Value:	216	Format:	U8	
Default Value:	216					
Format:	U8					

COLOR_PROCESSING_STATE - ACE State				
	7:0	B9		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">196</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 9 in PWL.</p>	Default Value:	196
Default Value:	196			
Format:	U8			
7	31:27	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		
	26:16	S1		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1.10</td> </tr> </table> <p>The value of the slope for point 1 in PWL. The default is 1024/1024.</p>	Format:	U1.10	
Format:	U1.10			
8	15:11	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		
	10:0	S0		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1.10</td> </tr> </table> <p>The value of the slope for point 0 in PWL. The default is 1024/1024.</p>	Format:	U1.10	
Format:	U1.10			
9	31:27	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		
	26:16	S3		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1.10</td> </tr> </table> <p>The value of the slope for point 3 in PWL. The default is 1024/1024.</p>	Format:	U1.10	
Format:	U1.10			
10	15:11	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		
	10:0	S2		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1.10</td> </tr> </table> <p>The value of the slope for point 2 in PWL. The default is 1024/1024.</p>	Format:	U1.10	
Format:	U1.10			
9	31:27	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		
	26:16	S5		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1.10</td> </tr> </table> <p>The value of the slope for point 5 in PWL. The default is 1024/1024.</p>	Format:	U1.10	
Format:	U1.10			
10	15:11	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		
	10:0	S4		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1.10</td> </tr> </table> <p>The value of the slope for point 4 in PWL. The default is 1024/1024.</p>	Format:	U1.10	
Format:	U1.10			
10	31:27	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			

COLOR_PROCESSING_STATE - ACE State				
	26:16	<p>S7</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U1.10</td> </tr> </table> <p>The value of the slope for point 7 in PWL. The default is 1024/1024.</p>	Format:	U1.10
	Format:	U1.10		
	15:11	<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			
10:0	<p>S6</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U1.10</td> </tr> </table> <p>The value of the slope for point 6 in PWL. The default is 1024/1024.</p>	Format:	U1.10	
Format:	U1.10			
11	31:27	<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		
	26:16	<p>S9</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U1.10</td> </tr> </table> <p>The value of the slope for point 9 in PWL. The default is 1024/1024.</p>	Format:	U1.10
	Format:	U1.10		
15:11	<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			
10:0	<p>S8</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U1.10</td> </tr> </table> <p>The value of the slope for point 8 in PWL. The default is 1024/1024.</p>	Format:	U1.10	
Format:	U1.10			
12	31:11	<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		
10:0	<p>S10</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U1.10</td> </tr> </table> <p>The value of the slope for point 10 in PWL. The default is 1024/1024.</p>	Format:	U1.10	
Format:	U1.10			

COLOR_PROCESSING_STATE - CSC State

COLOR_PROCESSING_STATE - CSC State		
Source:	BSpec	
Size (in bits):	288	
Default Value:	0x00002000, 0x00000000, 0x00000400, 0x00000000, 0x000004B4, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
This state structure contains the CSC state used by the color processing function. It corresponds to DW55..DW63 of the Color Processing State.		
DWord	Bit	Description
0	31:29	Reserved
		Format: MBZ
	28:16	C1
		Default Value: 0
		Format: S2.10 2's complement Transform coefficient
	15:3	C0
		Default Value: 1024
Format: S2.10 2's complement Transform coefficient		
2	YUV_IN	
	Default Value: 0	
	Format: YUV CSC input offset enable.	
1	YUV_OUT	
	Default Value: 0	
	Format: RGB CSC output offset enable.	
0	Transform Enable	
	Format: Enable	
1	31:26	Reserved
		Format: MBZ
	25:13	C3
Default Value: 0		
Format: S2.10 2's complement Transform coefficient.		

COLOR_PROCESSING_STATE - CSC State						
	12:0	C2 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>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					
2	31:26	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	25:13	C5 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>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>Default Value:</td> <td>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					
3	31:26	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	25:13	C7 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>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>Default Value:</td> <td>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					
4	31:13	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
12:0	C8 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>1204</td> </tr> <tr> <td>Format:</td> <td>S2.10 2's complement</td> </tr> </table> Transform coefficient.	Default Value:	1204	Format:	S2.10 2's complement	
Default Value:	1204					
Format:	S2.10 2's complement					
5	31:20	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
19:10	Offset out 1 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S9 2's complement</td> </tr> </table> Offset Out for Y/R.	Default Value:	0	Format:	S9 2's complement	
Default Value:	0					
Format:	S9 2's complement					

COLOR_PROCESSING_STATE - CSC State										
	9:0	Offset In 1 Default Value: 0 Format: S9 2's complement Offset in for Y/R.								
		Reserved Format: MBZ								
		19:10 Offset out 2 Default Value: 0 Format: S9 2's complement Offset out for U/G.								
6	9:0	Offset in 2 Default Value: 0 Format: S9 2's complement Offset in for U/G.								
		Reserved Format: MBZ								
		19:10 Offset out 3 Default Value: 0 Format: S9 2's complement Offset out for V/B.								
7	9:0	Offset in 3 Default Value: 0 Format: S9 2's complement Offset in for V/B.								
		Reserved Format: MBZ								
		19:10 Offset out 3 Default Value: 0 Format: S9 2's complement Offset out for V/B.								
8	31:17	Reserved Format: MBZ								
		Alpha from State Select Format: U1 Enumerated Type <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Alpha is taken from message</td> </tr> <tr> <td>1</td> <td></td> <td>Alpha is taken from state</td> </tr> </tbody> </table>	Value	Name	Description	0		Alpha is taken from message	1	
	Value	Name	Description							
	0		Alpha is taken from message							
1		Alpha is taken from state								
15:0	Color Pipe Alpha Format: U16									
	16 Alpha from State Select Format: U1 Enumerated Type <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Alpha is taken from message</td> </tr> <tr> <td>1</td> <td></td> <td>Alpha is taken from state</td> </tr> </tbody> </table>	Value	Name	Description	0		Alpha is taken from message	1		Alpha is taken from state
Value	Name	Description								
0		Alpha is taken from message								
1		Alpha is taken from state								

COLOR_PROCESSING_STATE - PROCAMP State

COLOR_PROCESSING_STATE - PROCAMP State		
Source:	BSpec	
Size (in bits):	64	
Default Value:	0x00020001, 0x01000000	
This state structure contains the PROCAMP state used by the color processing function. It corresponds to DW53..DW54 of the Color Processing State.		
DWord	Bit	Description
0	31:28	Reserved
		Format: MBZ
	27:17	Contrast
		Default Value: 1
		Format: U4.7 Contrast magnitude.
	16:13	Reserved
Format: MBZ		
12:1	Brightness	
	Default Value: 0	
	Format: S7.4 2's complement Brightness magnitude.	
0	PROCAMP Enable	
	Default Value: 1 Format: Enable	
1	31:16	Cos_c_s
		Default Value: 256 Format: S7.8 2's complement UV multiplication cosine factor.
	15:0	Sin_c_s
		Default Value: 0 Format: S7.8 2's complement UV multiplication sine factor.

COLOR_PROCESSING_STATE - STD/STE State

COLOR_PROCESSING_STATE - STD/STE State			
Source:	BSpec		
Size (in bits):	928		
Default Value:	0x9A6E39F0, 0x400C0000, 0x00001180, 0xFE2F2E00, 0x000000FF, 0x00140000, 0xD82E0000, 0x8285ECEC, 0x00008282, 0x00000000, 0x02117000, 0xA38FEC96, 0x00008CC8, 0x00000000, 0x01478000, 0x0007C300, 0x00000000, 0x00000000, 0x1C180000, 0x00000000, 0x00000000, 0x00000000, 0x0007CF80, 0x00000000, 0x00000000, 0x1C080000, 0x00000000, 0x00000000, 0x00000000		
This state structure contains the STD/STE state used by the color processing function.			
DWord	Bit	Description	
0	31:24	V_Mid	
		Default Value:	154
		Format:	U8
			Rectangle middle-point V coordinate
	23:16	U_Mid	
		Default Value:	110
		Format:	U8
			Rectangle middle-point U coordinate
	15:10	Hue Max	
		Default Value:	14
Format:		U6	
		Rectangle half width	
9:4	Sat Max		
	Default Value:	31	
	Format:	U6	
		Rectangle half length.	
3	Reserved		
	Format:	MBZ	
2	Output Control		
	Value	Name	
	0	Output Pixels [Default]	
1	Output STD Decisions		
1	STE Enable		
	Format:	Enable	
0	STD Enable		
	Format:	Enable	

COLOR_PROCESSING_STATE - STD/STE State		
1	31	Reserved Format: MBZ
	30:28	Diamond Margin Default Value: 4 Format: U3
		Diamond du Default Value: 0 Format: S6 2's complement Rhombus center shift in the sat-direction, relative to the rectangle center.
	20:18	HS Margin Default Value: 3 Format: U3
		Cos(α) Format: S0.7 2's Compliment The default is 79/128
	9:8	Reserved Format: MBZ
	7:0	Sin(α) Format: S0.7 2's Compliment The default is 101/128
	2	31:21
20:13		Diamond Alpha Format: U2.6 $1 / \tan(\beta)$ The default is 100/64
		Diamond Th Default Value: 35 Format: U6 Half length of the rhombus axis in the sat-direction.
6:0		Diamond dv Default Value: 0 Format: S6 2's complement
3	31:24	Y_point_3 Default Value: 254 Format: U8 Third point of the Y piecewise linear membership function.

COLOR_PROCESSING_STATE - STD/STE State										
	23:16	Y_point_2 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">47</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Second point of the Y piecewise linear membership function.	Default Value:	47	Format:	U8				
	Default Value:	47								
	Format:	U8								
	15:8	Y_point_1 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">46</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> First point of the Y piecewise linear membership function.	Default Value:	46	Format:	U8				
Default Value:	46									
Format:	U8									
7	VY_STD_Enable <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">Enable</td> </tr> </table> Enables STD in the VY subspace.	Format:	Enable							
Format:	Enable									
6:0	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ									
4	31:18	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ						
	Format:	MBZ								
	17:13	Y_Slope_2 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U2.3</td> </tr> </table> Slope between points Y3 and Y4. The default is 31/8.	Format:	U2.3						
	Format:	U2.3								
12:8	Y_Slope_1 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U2.3</td> </tr> </table> Slope between points Y1 and Y2. The default is 31/8.	Format:	U2.3							
Format:	U2.3									
7:0	Y_point_4 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">255</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Fourth point of the Y piecewise linear membership function	Default Value:	255	Format:	U8					
Default Value:	255									
Format:	U8									
5	31:16	INV_skin_types_margin <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U0.16</td> </tr> </table> $1/(2 * \text{Skin_types_margin})$ <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;">20</td> <td style="text-align: center;">[Default]</td> <td style="text-align: center;">Skin_Type_margin</td> </tr> </tbody> </table>	Format:	U0.16	Value	Name	Description	20	[Default]	Skin_Type_margin
	Format:	U0.16								
Value	Name	Description								
20	[Default]	Skin_Type_margin								
15:0	Inverse Margin VYL <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U0.16</td> </tr> </table> $1 / \text{Margin_VYL}$ The default is 3300/65536	Format:	U0.16							
Format:	U0.16									
6	31:24	P1L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">216</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Y Point 1 of the lower part of the detection PWLF.	Default Value:	216	Format:	U8				
		Default Value:	216							
Format:	U8									

COLOR_PROCESSING_STATE - STD/STE State						
	23:16	P0L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">46</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> <p>Y Point 0 of the lower part of the detection PWLF.</p>	Default Value:	46	Format:	U8
	Default Value:	46				
Format:	U8					
	15:0	Inverse Margin VYU <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">U0.16</td> </tr> </table> <p>1 / Margin_VYU The default is 1600/65536.</p>	Format:	U0.16		
Format:	U0.16					
7	31:24	B1L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">130</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> <p>V Bias 1 of the lower part of the detection PWLF.</p>	Default Value:	130	Format:	U8
	Default Value:	130				
	Format:	U8				
	23:16	B0L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">133</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> <p>V Bias 0 of the lower part of the detection PWLF.</p>	Default Value:	133	Format:	U8
Default Value:	133					
Format:	U8					
15:8	P3L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">236</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">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	P2L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">236</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">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					
8	31:27	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	26:16	S0L <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">S2.8 2's complement</td> </tr> </table> <p>Slope 0 of the lower part of the detection PWLF. The default is -5/256.</p>	Format:	S2.8 2's complement		
	Format:	S2.8 2's complement				
15:8	B3L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">130</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">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	B2L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">130</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">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					
9	31:22	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					

COLOR_PROCESSING_STATE - STD/STE State		
	21:11	S2L Format: S2.8 2's complement Slope 2 of the lower part of the detection PWLF. The default is 0/256.
	10:0	S1L Format: S2.8 2's complement Slope 1 of the lower part of the detection PWLF. The default is 0/256.
10	31:27	Reserved Format: MBZ
	26:19	P1U Default Value: 66 Format: U8 Y Point 1 of the upper part of the detection PWLF.
	18:11	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. The default is 0/256.
11	31:24	B1U Default Value: 163 Format: U8 V Bias 1 of the upper part of the detection PWLF.
	23:16	B0U Default Value: 143 Format: U8 V Bias 0 of the upper part of the detection PWLF.
	15:8	P3U Default Value: 236 Format: U8 Y Point 3 of the upper part of the detection PWLF.
	7:0	P2U Default Value: 150 Format: U8 Y Point 2 of the upper part of the detection PWLF.
12	31:27	Reserved Format: MBZ

COLOR_PROCESSING_STATE - STD/STE State									
	26:16	<p>S0U</p> <table border="1"> <tr> <td>Format:</td> <td>S2.8 2's complement</td> </tr> </table> <p>Slope 0 of the upper part of the detection PWLF. The default is 256/256.</p>	Format:	S2.8 2's complement					
	Format:	S2.8 2's complement							
	15:8	<p>B3U</p> <table border="1"> <tr> <td>Default Value:</td> <td>140</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 3 of the upper part of the detection PWLF.</p>	Default Value:	140	Format:	U8			
Default Value:	140								
Format:	U8								
7:0	<p>B2U</p> <table border="1"> <tr> <td>Default Value:</td> <td>200</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 2 of the upper part of the detection PWLF.</p>	Default Value:	200	Format:	U8				
Default Value:	200								
Format:	U8								
13	31:22	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
	21:11	<p>S2U</p> <table border="1"> <tr> <td>Format:</td> <td>S2.8 2's complement</td> </tr> </table> <p>Slope 2 of the upper part of the detection PWLF. The default is -179/256.</p>	Format:	S2.8 2's complement					
Format:	S2.8 2's complement								
10:0	<p>S1U</p> <table border="1"> <tr> <td>Format:</td> <td>S2.8 2's complement</td> </tr> </table> <p>Slope 1 of the upper part of the detection PWLF. The default is -113/256.</p>	Format:	S2.8 2's complement						
Format:	S2.8 2's complement								
14	31:28	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
	Format:	MBZ							
	27:20	<p>Skin Types Margin</p> <table border="1"> <tr> <td>Default Value:</td> <td>20</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Skin types Y margin.</p>	Default Value:	20	Format:	U8			
	Default Value:	20							
	Format:	U8							
19:12	<p>Skin Types Thresh</p> <table border="1"> <tr> <td>Default Value:</td> <td>120</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Skin types Y threshold.</p>	Default Value:	120	Format:	U8				
Default Value:	120								
Format:	U8								
11	<p>Skin Type Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Treat differently bright and dark skin types.</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>Disable</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0	[Default]	Disable
Format:	Enable								
Value	Name	Description							
0	[Default]	Disable							
10:0	<p>S3U</p> <table border="1"> <tr> <td>Format:</td> <td>S2.8 2's complement</td> </tr> </table> <p>Slope 3 of the upper part of the detection PWLF. The default is 0/256.</p>	Format:	S2.8 2's complement						
Format:	S2.8 2's complement								

COLOR_PROCESSING_STATE - STD/STE State		
15	31	Reserved Format: MBZ
	30:21	SATB1 Format: S7.2 2's complement First bias for the saturation PWLF (bright skin). The default is -8/4.
	20:14	SATP3 Default Value: 31 Format: S6 2's complement Third point for the saturation PWLF (bright skin).
	13:7	SATP2 Default Value: 6 Format: S6 2's complement Second point for the saturation PWLF (bright skin).
	6:0	SATP1 Format: S6 2's complement First point for the saturation PWLF (bright skin). The default is -6.
16	31	Reserved Format: MBZ
	30:20	SATS0 Format: U3.8 Zeroth slope for the saturation PWLF (bright skin). The default is 297/256.
	19:10	SATB3 Format: S7.2 2's complement Third bias for the saturation PWLF (bright skin). The default is 124/4.
	9:0	SATB2 Format: S7.2 2's complement Second bias for the saturation PWLF (bright skin). The default is 8/4.
17	31:22	Reserved Format: MBZ
	21:11	SATS2 Format: U3.8 Second slope for the saturation PWLF (bright skin). The default is 297/256.
	10:0	SATS1 Format: U3.8 First slope for the saturation PWLF (bright skin). The default is 85/256.

COLOR_PROCESSING_STATE - STD/STE State						
18	31:25	HUEP3 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>14</td> </tr> <tr> <td>Format:</td> <td>S6 2's complement</td> </tr> </table> Third point for the hue PWLF (bright skin)	Default Value:	14	Format:	S6 2's complement
	Default Value:	14				
	Format:	S6 2's complement				
	24:18	HUEP2 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>6</td> </tr> <tr> <td>Format:</td> <td>S6 2's complement</td> </tr> </table> Second point for the hue PWLF (bright skin)	Default Value:	6	Format:	S6 2's complement
Default Value:	6					
Format:	S6 2's complement					
17:11	HUEP1 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S6 2's complement</td> </tr> </table> First point for the hue PWLF (bright skin). The default is -6.	Format:	S6 2's complement			
Format:	S6 2's complement					
10:0	SATS3 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table> Thrid slope for the saturation PWLF (bright skin). The default is 256/256.	Format:	U3.8			
Format:	U3.8					
19	31:30	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	29:20	HUEB3 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S7.2 2's complement</td> </tr> </table> Third bias for the hue PWLF (bright skin). The default is 56/4.	Format:	S7.2 2's complement		
	Format:	S7.2 2's complement				
19:10	HUEB2 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S7.2 2's complement</td> </tr> </table> Second bias for the hue PWLF (bright skin). The default is 8/4.	Format:	S7.2 2's complement			
Format:	S7.2 2's complement					
9:0	HUEB1 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S7.2 2's complement</td> </tr> </table> First bias for the hue PWLF (bright skin). The default is -8/4.	Format:	S7.2 2's complement			
Format:	S7.2 2's complement					
20	31:22	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	21:11	HUES1 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table> First slope for the hue PWLF (bright skin) The default is 85/256.	Format:	U3.8		
Format:	U3.8					
10:0	HUES0 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table> Zeroth slope for the hue PWLF (bright skin) The default is 384/256.	Format:	U3.8			
Format:	U3.8					
21	31:22	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					

COLOR_PROCESSING_STATE - STD/STE State		
	21:11	HUES3 Format: U3.8 Third slope for the hue PWLF (bright skin) The default is 256/256.
	10:0	HUES2 Format: U3.8 Second slope for the hue PWLF (bright skin) The default is 384/256.
22	31	Reserved Format: MBZ
	30:21	SATB1_DARK Format: S7.2 2's complement First bias for the saturation PWLF (dark skin) The default is 0/4.
	20:14	SATP3_DARK Default Value: 31 Format: S6 2's complement Third point for the saturation PWLF (dark skin)
	13:7	SATP2_DARK Default Value: 31 Format: S6 2's complement Second point for the saturation PWLF (dark skin)
	6:0	SATP1_DARK Format: S6 2's complement First point for the saturation PWLF (dark skin). The default is -11.
	23	31
30:20		SATS0_DARK Format: U3.8 Zeroth slope for the saturation PWLF (dark skin). The default is 397/256.
19:10		SATB3_DARK Format: S7.2 2's complement Third bias for the saturation PWLF (dark skin). The default is 124/4.
9:0		SATB2_DARK Format: S7.2 2's complement Second bias for the saturation PWLF (dark skin). The default is 124/4.
24	31:22	Reserved Format: MBZ
	21:11	SATS2_DARK Format: U3.8 Second slope for the saturation PWLF (dark skin). The default is 256/256.

COLOR_PROCESSING_STATE - STD/STE State		
	10:0	SATS1_DARK
		Format: U3.8
		First slope for the saturation PWLF (dark skin). The default is 189/256.
25	31:25	HUEP3_DARK
		Default Value: 14
		Format: S6 2's complement
		Third point for the hue PWLF (dark skin).
	24:18	HUEP2_DARK
		Default Value: 2
		Format: S6 2's complement
		Third point for the hue PWLF (dark skin).
	17:11	HUEP1_DARK
		Default Value: 0
		Format: S6 2's complement
		Third point for the hue PWLF (dark skin).
10:0	SATS3_DARK	
	Format: U3.8	
	Third slope for the saturation PWLF (dark skin). The default is 256/256.	
26	31:30	Reserved
		Format: MBZ
	29:20	HUEB3_DARK
		Format: S7.2 2's complement Third bias for the hue PWLF (dark skin). The default is 56/4.
	19:10	HUEB2_DARK
		Format: S7.2 2's complement Second bias for the hue PWLF (dark skin). The default is 0/4.
	9:0	HUEB1_DARK
		Format: S7.2 2's complement First bias for the hue PWLF (dark skin). The default is 0/4.
27	31:22	Reserved
		Format: MBZ
	21:11	HUES1_DARK
		Format: U3.8 First slope for the hue PWLF (dark skin). The default is 0/256.
	10:0	HUES0_DARK
		Format: U3.8 Zeroth slope for the hue PWLF (dark skin). The default is 256/256.



COLOR_PROCESSING_STATE - STD/STE State		
28	31:22	Reserved
		Format: MBZ
	21:11	HUES3_DARK
		Format: U3.8
		Third slope for the hue PWLF (dark skin). The default is 256/256.
	10:0	HUES2_DARK
Format: U3.8		
Second slope for the hue PWLF (dark skin). The default is 299/256.		

COLOR_PROCESSING_STATE - TCC State

COLOR_PROCESSING_STATE - TCC State			
Source:	BSpec		
Size (in bits):	352		
Default Value:	0xDCDCDC00, 0xDCDCDC00, 0x1E34CC91, 0x3E3CCE91, 0x02E80195, 0x0197046B, 0x01790174, 0x00096000, 0x00000000, 0x03030000, 0x009201C0		
This state structure contains the TCC state used by the color processing function. It corresponds to DW42..DW52 of the Color Processing State.			
DWord	Bit	Description	
0	31:24	SatFactor3	
		Default Value:	220
		Format:	U1.7
		The saturation factor for yellow.	
	23:16	SatFactor2	
		Default Value:	220
		Format:	U1.7
		The saturation factor for red.	
	15:8	SatFactor1	
		Default Value:	220
Format:		U1.7	
The saturation factor for magenta.			
7	TCC Enable		
Format:		Enable	
6:0	Reserved		
Format:		MBZ	
1	31:24	SatFactor6	
		Default Value:	220
		Format:	U1.7
		The saturation factor for blue.	
	23:16	SatFactor5	
		Default Value:	220
		Format:	U1.7
		The saturation factor for cyan.	
	15:8	SatFactor4	
Default Value:		220	
Format:		U1.7	
The saturation factor for green.			

COLOR_PROCESSING_STATE - TCC State		
	7:0	Reserved Format: MBZ
2	31:30	Reserved Format: MBZ
	29:20	Base Color 3 Default Value: 483 Format: U10
		Base Color 2 Default Value: 307 Format: U10
	9:0	Base Color 1 Default Value: 145 Format: U10
	3	31:30
29:20		Base Color 6 Default Value: 995 Format: U10
		Base Color 5 Default Value: 819 Format: U10
9:0		Base Color 4 Default Value: 657 Format: U10
4	31:16	Color Transit Slope 23 Default Value: 744 Format: U0.16 The calculation result of $1 / (BC3 - BC2)$ [1/62]
		Color Transit Slope 12 Default Value: 405 Format: U0.16 The calculation result of $1 / (BC2 - BC1)$ [1/57]
5	31:16	Color Transit Slope 45 Default Value: 407 Format: U0.16 The calculation result of $1 / (BC5 - BC4)$ [1/57]

COLOR_PROCESSING_STATE - TCC State						
	15:0	Color Transit Slope 34				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">1131</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>The calculation result of 1 / (BC4 - BC3) [1/61]</p>	Default Value:	1131	Format:	U0.16
Default Value:	1131					
Format:	U0.16					
6	31:16	Color Transit Slope 61				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">377</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>The calculation result of 1 / (BC1 - BC6) [1/62]</p>	Default Value:	377	Format:	U0.16
	Default Value:	377				
	Format:	U0.16				
15:0	Color Transit Slope 56					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">372</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>The calculation result of 1 / (BC6 - BC5) [1/62]</p>	Default Value:	372	Format:	U0.16	
Default Value:	372					
Format:	U0.16					
7	31:22	Color Bias 3				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">0</td> </tr> <tr> <td>Format:</td> <td>U2.8</td> </tr> </table> <p>Color bias for BaseColor3.</p>	Default Value:	0	Format:	U2.8
	Default Value:	0				
	Format:	U2.8				
21:12	Color Bias 2					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">150</td> </tr> <tr> <td>Format:</td> <td>U2.8</td> </tr> </table> <p>Color bias for BaseColor2.</p>	Default Value:	150	Format:	U2.8	
Default Value:	150					
Format:	U2.8					
11:2	Color Bias 1					
1:0	Reserved					
8	31:22	Color Bias 6				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">0</td> </tr> <tr> <td>Format:</td> <td>U2.8</td> </tr> </table> <p>Color bias for BaseColor6.</p>	Default Value:	0	Format:	U2.8
	Default Value:	0				
	Format:	U2.8				
	21:12	Color Bias 5				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">0</td> </tr> <tr> <td>Format:</td> <td>U2.8</td> </tr> </table> <p>Color bias for BaseColor5.</p>	Default Value:	0	Format:	U2.8
Default Value:	0					
Format:	U2.8					
11:2	ColorBias4					
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">0</td> </tr> <tr> <td>Format:</td> <td>U2.8</td> </tr> </table> <p>Color bias for BaseColor4.</p>	Default Value:	0	Format:	U2.8
Default Value:	0					
Format:	U2.8					
		<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					

COLOR_PROCESSING_STATE - TCC State						
	1:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					
9	31	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	30:24	<p>UV Threshold</p> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>3</td> </tr> <tr> <td>Format:</td> <td>U7</td> </tr> </table> <p>Low UV threshold.</p>	Default Value:	3	Format:	U7
	Default Value:	3				
	Format:	U7				
	23:19	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	18:16	<p>UV Threshold Bits</p> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>3</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Low UV transition width bits.</p>	Default Value:	3	Format:	U3
	Default Value:	3				
Format:	U3					
15:13	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
12:8	<p>STE Threshold</p> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Skin tone pixels enhancement threshold.</p>	Default Value:	0	Format:	U5	
Default Value:	0					
Format:	U5					
7:3	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
2:0	<p>STE Slope Bits</p> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Skin tone pixels enhancement slope bits.</p>	Default Value:	0	Format:	U3	
Default Value:	0					
Format:	U3					
10	31:16	<p>Inverse UVMax Color</p> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>146</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>1 / UVMaxColor. Used for the SFs2 calculation.</p>	Default Value:	146	Format:	U0.16
	Default Value:	146				
	Format:	U0.16				
15:9	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
8:0	<p>UVMax Color</p> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>448</td> </tr> <tr> <td>Format:</td> <td>U9</td> </tr> </table> <p>The maximum absolute value of the legal UV pixels. Used for the SFs2 calculation.</p>	Default Value:	448	Format:	U9	
Default Value:	448					
Format:	U9					

Color Calculator State Pointer Message Header Control

MHC_RT_CCSP - Color Calculator State Pointer Message Header Control				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:6	<p>Color Calculator State Pointer</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GeneralStateOffset[31:6]</td> </tr> </table> <p>Specifies the 64-byte aligned point to the color calculator state. This pointer is relative to the General State Base Address.</p>	Format:	GeneralStateOffset[31:6]
	Format:	GeneralStateOffset[31:6]		
5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore	
Format:	Ignore			

Color Code Message Header Control

MHC_RT_CC - Color Code Message Header Control				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:10	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
	Format:	Ignore		
	9:8	<p>Color Code</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U2</td> </tr> </table> <p>This ID is assigned by the Windower unit and is used to track synchronizing events. Reserved for HW implementation use</p>	Format:	U2
Format:	U2			
7:0	<p>FFTID</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U8</td> </tr> </table> <p>This ID is assigned by the fixed function unit and is a unique identifier for the thread. It is used to free up resources used by the thread upon thread completion.</p>	Format:	U8	
Format:	U8			

Context Descriptor Format

Context Descriptor Format							
Source:	BSpec						
Size (in bits):	64						
Default Value:	0x00000000, 0x00000000						
This is the format of context descriptors which make up submitted execlists.							
DWord	Bit	Description					
0	63:32	<p>Context ID</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; background-color: #e6f2ff;">Description</th> </tr> </thead> <tbody> <tr> <td>Context ID is a unique field assigned by GFX driver when a new context is created by which it is identified across all hierarchies of SW and HW. <ul style="list-style-type: none"> Context ID is used for semaphore signaling by hardware and software. Context ID matching is used by hardware to detect Lite Restore. Context ID is used by hardware for page fault reporting and response with IOMMU. Context switch reason and the associated Context ID are reported to Context Switch Status Buffer by hardware on a context switch. </td> </tr> <tr> <td>Context ID[31:0] (bits[63:32] of the context descriptor) are used for comparing during lite restore, semaphore signaling and context specific OA enabling.</td> </tr> <tr> <td>Context ID which is a 32 bit field is further divided in to three segments described below: <ul style="list-style-type: none"> Bits[63:55] (Bits 31:23 of Context ID) is referred to as GroupID. GroupID+PASID combination of a context must be a unique identifier for contexts that are active in the system. The definition of active context is listed as: <ul style="list-style-type: none"> Any Context that is already submitted to h/w or already running in h/w. Any Context that hit page faults, was preempted (didn't run to context complete), and is waiting to be resubmitted pending IOMMU "last in group" response. Any Context that has experienced reset but not all faults are responded to. Bit[54] (Bit 22 of Context ID) – MBZ for SW programming; this bit is used by hardware to distinguish between F&H vs F&S page requests and response messages to and from IOMMU. This bit is used by hardware on receiving page response to properly manage the page fault counters Bit[53] (Bit 21 of Context ID) – MBZ from SW programming, is reserved for future hardware use. Bits[52:32] (Bits 20:0 of Context ID) are for software use-only and must be unique field assigned by GFX driver when a new context is created. </td> </tr> <tr> <td>Context ID is reported by hardware to OABUFFER along with the performance statistics counters, Context ID is used for filtering the statistics on per context basis.</td> </tr> </tbody> </table>	Description	Context ID is a unique field assigned by GFX driver when a new context is created by which it is identified across all hierarchies of SW and HW. <ul style="list-style-type: none"> Context ID is used for semaphore signaling by hardware and software. Context ID matching is used by hardware to detect Lite Restore. Context ID is used by hardware for page fault reporting and response with IOMMU. Context switch reason and the associated Context ID are reported to Context Switch Status Buffer by hardware on a context switch. 	Context ID[31:0] (bits[63:32] of the context descriptor) are used for comparing during lite restore, semaphore signaling and context specific OA enabling.	Context ID which is a 32 bit field is further divided in to three segments described below: <ul style="list-style-type: none"> Bits[63:55] (Bits 31:23 of Context ID) is referred to as GroupID. GroupID+PASID combination of a context must be a unique identifier for contexts that are active in the system. The definition of active context is listed as: <ul style="list-style-type: none"> Any Context that is already submitted to h/w or already running in h/w. Any Context that hit page faults, was preempted (didn't run to context complete), and is waiting to be resubmitted pending IOMMU "last in group" response. Any Context that has experienced reset but not all faults are responded to. Bit[54] (Bit 22 of Context ID) – MBZ for SW programming; this bit is used by hardware to distinguish between F&H vs F&S page requests and response messages to and from IOMMU. This bit is used by hardware on receiving page response to properly manage the page fault counters Bit[53] (Bit 21 of Context ID) – MBZ from SW programming, is reserved for future hardware use. Bits[52:32] (Bits 20:0 of Context ID) are for software use-only and must be unique field assigned by GFX driver when a new context is created. 	Context ID is reported by hardware to OABUFFER along with the performance statistics counters, Context ID is used for filtering the statistics on per context basis.
Description							
Context ID is a unique field assigned by GFX driver when a new context is created by which it is identified across all hierarchies of SW and HW. <ul style="list-style-type: none"> Context ID is used for semaphore signaling by hardware and software. Context ID matching is used by hardware to detect Lite Restore. Context ID is used by hardware for page fault reporting and response with IOMMU. Context switch reason and the associated Context ID are reported to Context Switch Status Buffer by hardware on a context switch. 							
Context ID[31:0] (bits[63:32] of the context descriptor) are used for comparing during lite restore, semaphore signaling and context specific OA enabling.							
Context ID which is a 32 bit field is further divided in to three segments described below: <ul style="list-style-type: none"> Bits[63:55] (Bits 31:23 of Context ID) is referred to as GroupID. GroupID+PASID combination of a context must be a unique identifier for contexts that are active in the system. The definition of active context is listed as: <ul style="list-style-type: none"> Any Context that is already submitted to h/w or already running in h/w. Any Context that hit page faults, was preempted (didn't run to context complete), and is waiting to be resubmitted pending IOMMU "last in group" response. Any Context that has experienced reset but not all faults are responded to. Bit[54] (Bit 22 of Context ID) – MBZ for SW programming; this bit is used by hardware to distinguish between F&H vs F&S page requests and response messages to and from IOMMU. This bit is used by hardware on receiving page response to properly manage the page fault counters Bit[53] (Bit 21 of Context ID) – MBZ from SW programming, is reserved for future hardware use. Bits[52:32] (Bits 20:0 of Context ID) are for software use-only and must be unique field assigned by GFX driver when a new context is created. 							
Context ID is reported by hardware to OABUFFER along with the performance statistics counters, Context ID is used for filtering the statistics on per context basis.							

Context Descriptor Format

31:12	Logical Ring Context Address (LRCA)	
	Format:	GraphicsAddress[31:12]
	This field contains the 4 KB-aligned address of the Logical Ring Context associated with this execlist element. LRCA must be always programmed in GGTT memory.	
11:9	Reserved	
	Format:	MBZ
8	Privilege Access	
	This field when set indicates PPGTT enabled in legacy context mode. In advanced context mode this field is reserved and must be zero.	
7:6	Fault Handling	
	Source:	CommandStreamer
	Value	Name
	Description	
0h	Fault and Hang	Fault model is not supported and fault occurrence is treated as catastrophic. GAM indicates Fault Error to Command streamer. Fault Error interrupt is reported to scheduler. Command Streamer will not initiate context switch on occurrence of Fault Error.
1h	Reserved	Reserved
2h	Fault and Stream	In this mode of operation faults are allowed on EU memory surfaces. Page Walker will directly work with memory page handler to fix the faults on the fly for these surfaces. Command streamer is not aware of the fault service being done by page walker and goes with its normal execution rules for context switch. On completion of flush during context switch CS explicitly requests acknowledge message to Page Walker before proceeding further. Page Walker acknowledges Command Streamer once it is done on a clean boundary. Page Walker asserts Fault Error on occurrence of non recoverable fault or access violations (Command Streamer access, VFunit access, etc) to Command Streamer; this is the same as Fault and Hang behavior.
3h	Reserved	Reserved
	Programming Notes	
	When execlist mode is set to "Legacy Context mode" Fault Handling mode must be set to "Fault and Hang." For proper programming for Page Fault modes, refer to memory interface section of the Bspec for the corresponding generation.	
5	Reserved	
	Format:	MBZ

Context Descriptor Format

4:3	Addressing Mode & Legacy Context	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 65%;">Format:</td> <td style="width: 35%; text-align: center;">U2</td> </tr> </table> <p>Legacy context set indicates GPU is operating in legacy context mode of operation and doesn't support any SVM features. Legacy context reset indicates GPU is operating in advanced context mode of operation and support SVM features. Based on the Context mode set Addressing mode is interpreted appropriately. The table below summarizes the combinations supported.</p> <p>GFX engine always uses 32b virtual addressing mode when translated using GGTT irrespective of below options.</p> <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>00b</td> <td>Advanced Context with no A/D support</td> <td>GPU is enabled for advanced context mode and supports SVM features. GPU DOESN'T support Access and Dirty bit management in page tables. GPU supports 64b(48bit canonical) PPGTT graphics virtual addressing. PDP0_DESCRIPTOR contains the PASID (process address space identifier) and other PDP Descriptors are ignored.</td> </tr> <tr> <td>01b</td> <td>Legacy Context with no 64 bit VA support</td> <td>GPU is enabled for legacy context mode of operation and DOESN'T support any SVM features. GPU supports 32b PPGTT graphics virtual addressing. PDP*__DESCRIPTOR contains the base address to 4GB of memory space supported.</td> </tr> <tr> <td>10b</td> <td>Advanced Context with A/D support</td> <td>GPU is enabled for advanced context mode and supports SVM features. GPU DOES support Access and Dirty bit management in page tables. GPU supports 64b (48bit canonical) PPGTT graphics virtual addressing. PDP0_DESCRIPTOR contains the PASID (process address space identifier) and other PDP Descriptors are ignored.</td> </tr> <tr> <td>11b</td> <td>Legacy Context with 64 bit VA support</td> <td>GPU is enabled for legacy context mode of operation and DOESN'T support any SVM features. GPU supports 64b (48bit canonical) PPGTT graphics virtual addressing and PDP0_DESCRIPTOR contains the base address to PML4 and other PDP Descriptors are ignored.</td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	00b	Advanced Context with no A/D support	GPU is enabled for advanced context mode and supports SVM features. GPU DOESN'T support Access and Dirty bit management in page tables. GPU supports 64b(48bit canonical) PPGTT graphics virtual addressing. PDP0_DESCRIPTOR contains the PASID (process address space identifier) and other PDP Descriptors are ignored.	01b	Legacy Context with no 64 bit VA support	GPU is enabled for legacy context mode of operation and DOESN'T support any SVM features. GPU supports 32b PPGTT graphics virtual addressing. PDP*__DESCRIPTOR contains the base address to 4GB of memory space supported.	10b	Advanced Context with A/D support	GPU is enabled for advanced context mode and supports SVM features. GPU DOES support Access and Dirty bit management in page tables. GPU supports 64b (48bit canonical) PPGTT graphics virtual addressing. PDP0_DESCRIPTOR contains the PASID (process address space identifier) and other PDP Descriptors are ignored.	11b	Legacy Context with 64 bit VA support	GPU is enabled for legacy context mode of operation and DOESN'T support any SVM features. GPU supports 64b (48bit canonical) PPGTT graphics virtual addressing and PDP0_DESCRIPTOR contains the base address to PML4 and other PDP Descriptors are ignored.
Format:	U2																		
Value	Name	Description																	
00b	Advanced Context with no A/D support	GPU is enabled for advanced context mode and supports SVM features. GPU DOESN'T support Access and Dirty bit management in page tables. GPU supports 64b(48bit canonical) PPGTT graphics virtual addressing. PDP0_DESCRIPTOR contains the PASID (process address space identifier) and other PDP Descriptors are ignored.																	
01b	Legacy Context with no 64 bit VA support	GPU is enabled for legacy context mode of operation and DOESN'T support any SVM features. GPU supports 32b PPGTT graphics virtual addressing. PDP*__DESCRIPTOR contains the base address to 4GB of memory space supported.																	
10b	Advanced Context with A/D support	GPU is enabled for advanced context mode and supports SVM features. GPU DOES support Access and Dirty bit management in page tables. GPU supports 64b (48bit canonical) PPGTT graphics virtual addressing. PDP0_DESCRIPTOR contains the PASID (process address space identifier) and other PDP Descriptors are ignored.																	
11b	Legacy Context with 64 bit VA support	GPU is enabled for legacy context mode of operation and DOESN'T support any SVM features. GPU supports 64b (48bit canonical) PPGTT graphics virtual addressing and PDP0_DESCRIPTOR contains the base address to PML4 and other PDP Descriptors are ignored.																	
2	Force Restore	<p>Setting this bit will force a context restore operation when switching to this context even if the LRCA in the CCID register (normally the LRCA of the last context from the prior execlist) matches this one.</p> <p>Note that it is legal (and likely desirable) for the Render Context Restore Inhibit bit (part of the CTXT_SR_CTL register) in the context image being restored to also be set. The "ring" context is being forced to be restored from a newly initialized context despite a possible LRCA match. However, the render context for such a newly initialized context will likely be uninitialized and so should not be restored.</p>																	
1	Force PD Restore	<p>Setting this bit will cause the on-chip page directory to be reloaded from the PD image in memory even on an LRCA match. No other operations of context restore will occur on an LRCA match, however. Software should set this bit if it has updated a context's page directory and wants the context to begin using the new page directory without having to switch away from it (to another context) and back again. Setting this bit will have no effect if Force Restore is also set; a complete context restore (including the PD) will be performed.</p>																	

Context Descriptor Format

	0	<p>Valid</p> <p>Set if this register holds a valid context descriptor. SW should set this bit in the Element registers that it has set up to contain valid context descriptors. Any execlist elements that are not used in a submitted execlist must have this bit clear.</p>
--	---	--

Context Status

Context Status																								
Source:	BSpec																							
Size (in bits):	64																							
Default Value:	0x00000000, 0x00000000																							
DWord	Bit	Description																						
0	63:32	Context ID Format: U32																						
	31:30	Reserved Format: MBZ																						
	29	Reserved Format: MBZ																						
	28	Reserved																						
	27:25	Reserved Format: MBZ																						
	24:20	Reserved Format: MBZ																						
	19:16	Display Plane This indicates the display plane for which Wait on Scanline/V-Blank/Sync Flip has been executed leading to context switch. This field is only valid when one of the "Wait on Scanline" or "Wait on Vblnak" or "Wait on sync Flip" is set. <table border="1" data-bbox="332 1222 1469 1722"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved (Look at field 14:12)</td> </tr> <tr> <td>1h</td> <td>Reserved</td> </tr> <tr> <td>2h</td> <td>Reserved</td> </tr> <tr> <td>3h</td> <td>Display Plane-7</td> </tr> <tr> <td>4h</td> <td>Display Plane-8</td> </tr> <tr> <td>5h</td> <td>Display Plane-9</td> </tr> <tr> <td>6h</td> <td>Display Plane-10</td> </tr> <tr> <td>7h</td> <td>Display Plane-11</td> </tr> <tr> <td>8h</td> <td>Display Plane-12</td> </tr> <tr> <td>[9h, Fh]</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	0h	Reserved (Look at field 14:12)	1h	Reserved	2h	Reserved	3h	Display Plane-7	4h	Display Plane-8	5h	Display Plane-9	6h	Display Plane-10	7h	Display Plane-11	8h	Display Plane-12	[9h, Fh]	Reserved
	Value	Name																						
	0h	Reserved (Look at field 14:12)																						
	1h	Reserved																						
2h	Reserved																							
3h	Display Plane-7																							
4h	Display Plane-8																							
5h	Display Plane-9																							
6h	Display Plane-10																							
7h	Display Plane-11																							
8h	Display Plane-12																							
[9h, Fh]	Reserved																							
15	Lite Restore Format: Enable This bit is only valid only when Preempted bit is set. When set, this bit indicates that a given context got preempted with the same context resulting in Lite Restore in HW.																							

Context Status

14:12	<p>Display Plane Additional</p> <p>This indicates the display plane for which Wait on Scanline/V-Blank/Sync Flip has been executed leading to context switch. This field is only valid when one of the "Wait on Scanline" or "Wait on Vblnak" or "Wait on sync Flip" is set. (Future - could remove the Sprites and move to bits 19:16)</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;">Exists If</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>Display Plane-1</td> <td>[Wait on V-blank]==0</td> </tr> <tr> <td style="text-align: center;">0h</td> <td>Display Pipe-A</td> <td>[Wait on V-blank]==1</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>Display Plane-2</td> <td>[Wait on V-blank]==0</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>Display Pipe-B</td> <td>[Wait on V-blank]==1</td> </tr> <tr> <td style="text-align: center;">2h</td> <td>Display Plane-3</td> <td>[Wait on V-blank]==0</td> </tr> <tr> <td style="text-align: center;">2h</td> <td>Display Pipe-C</td> <td>[Wait on V-blank]==1</td> </tr> <tr> <td style="text-align: center;">3h</td> <td>Display Plane-4</td> <td></td> </tr> <tr> <td style="text-align: center;">4h</td> <td>Display Plane-5</td> <td></td> </tr> <tr> <td style="text-align: center;">5h</td> <td>Display Plane-6</td> <td></td> </tr> </tbody> </table>		Value	Name	Exists If	0h	Display Plane-1	[Wait on V-blank]==0	0h	Display Pipe-A	[Wait on V-blank]==1	1h	Display Plane-2	[Wait on V-blank]==0	1h	Display Pipe-B	[Wait on V-blank]==1	2h	Display Plane-3	[Wait on V-blank]==0	2h	Display Pipe-C	[Wait on V-blank]==1	3h	Display Plane-4		4h	Display Plane-5		5h	Display Plane-6	
Value	Name	Exists If																														
0h	Display Plane-1	[Wait on V-blank]==0																														
0h	Display Pipe-A	[Wait on V-blank]==1																														
1h	Display Plane-2	[Wait on V-blank]==0																														
1h	Display Pipe-B	[Wait on V-blank]==1																														
2h	Display Plane-3	[Wait on V-blank]==0																														
2h	Display Pipe-C	[Wait on V-blank]==1																														
3h	Display Plane-4																															
4h	Display Plane-5																															
5h	Display Plane-6																															
11	<p>Semaphore Wait 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;">0h</td> <td>Signal Mode</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>Poll Mode</td> </tr> </tbody> </table>		Value	Name	0h	Signal Mode	1h	Poll Mode																								
Value	Name																															
0h	Signal Mode																															
1h	Poll Mode																															
10:9	<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																															
8	Wait on Scanline																															
7	Wait on Semaphore																															
6	Wait on V-blank																															
5	Wait on Sync Flip																															
4	<p>Context Complete</p> <p>Element is completely processed (Head eqv to Tail) and resulted in a context switch.</p>																															
3	<p>ACTIVE to IDLE</p> <p>Following this context switch there is no active element available in HW to execute</p>																															
2	<p>Element Switch</p> <p>Context Switch happened from first element in the current execlist to the second element of the same execlist</p>																															
1	<p>Preempted</p> <p>Submission of a new execlist has resulted in context switch. The switch is from element in current execlist to element in pending execlist</p>																															
0	<p>IDLE to ACTIVE</p> <p>Execlist submitted when HW is IDLE. When this bit is set rest of the fields in CSQ are not valid.</p>																															

CSC COEFFICIENT FORMAT

CSC COEFFICIENT FORMAT																										
Source:	BSpec																									
Size (in bits):	16																									
Default Value:	0x00000000																									
Coefficients for the CSC are stored in sign-exponent-mantissa format. Two CSC coefficients are stored in each dword, the table below show the data packing in each dword.																										
DWord	Bit	Description																								
0	15	Sign																								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Positive</td> </tr> <tr> <td>1b</td> <td>Negative</td> </tr> </tbody> </table>	Value	Name	0b	Positive	1b	Negative																		
		Value	Name																							
	0b	Positive																								
	1b	Negative																								
	14:12	Exponent_bits Represented as $2^{(-n)}$																								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>110b</td> <td>4</td> <td>4 or mantissa is bb.bbbbbbb</td> </tr> <tr> <td>111b</td> <td>2</td> <td>2 or mantissa is b.bbbbbbb</td> </tr> <tr> <td>000b</td> <td>1</td> <td>1 or mantissa is 0.bbbbbbb</td> </tr> <tr> <td>001b</td> <td>0.5</td> <td>0.5 or mantissa is 0.0bbbbbb</td> </tr> <tr> <td>010b</td> <td>0.25</td> <td>0.25 or mantissa is 0.00bbbbbb</td> </tr> <tr> <td>011b</td> <td>0.125</td> <td>0.125 or mantissa is 0.000bbbbbb</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	110b	4	4 or mantissa is bb.bbbbbbb	111b	2	2 or mantissa is b.bbbbbbb	000b	1	1 or mantissa is 0.bbbbbbb	001b	0.5	0.5 or mantissa is 0.0bbbbbb	010b	0.25	0.25 or mantissa is 0.00bbbbbb	011b	0.125	0.125 or mantissa is 0.000bbbbbb	Others	Reserved	Reserved
		Value	Name	Description																						
		110b	4	4 or mantissa is bb.bbbbbbb																						
		111b	2	2 or mantissa is b.bbbbbbb																						
000b		1	1 or mantissa is 0.bbbbbbb																							
001b		0.5	0.5 or mantissa is 0.0bbbbbb																							
010b		0.25	0.25 or mantissa is 0.00bbbbbb																							
011b	0.125	0.125 or mantissa is 0.000bbbbbb																								
Others	Reserved	Reserved																								
11:3	Mantissa																									
2:0	Reserved																									

Data Port 0 Message Types

MT_DP0 - Data Port 0 Message Types																																						
Source:	DataPort 0																																					
Size (in bits):	5																																					
Default Value:	0x00000000																																					
<p>Lists all the Message Types in a Data Port 0 Message Descriptor [18:14]. The Legacy messages are encoded in Data Port 0 with Bit 18 set to zero. The Message Header is optional for many (but not all) of these operations. The Scratch Block messages are encoded in Data Port 0 with Bit 18 set to one. A Message Header is required.</p>																																						
DWord	Bit	Description																																				
0	4	Legacy DAP-DC Message																																				
		Format: Enumeration																																				
		Legacy Message																																				
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No [Default]</td> <td>Legacy DAP-DC Message</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td>Scratch Block Message, descriptor uses different Message Type encoding</td> </tr> </tbody> </table>	Value	Name	Description	0h	No [Default]	Legacy DAP-DC Message	1h	Reserved	Scratch Block Message, descriptor uses different Message Type encoding																											
	Value	Name	Description																																			
	0h	No [Default]	Legacy DAP-DC Message																																			
	1h	Reserved	Scratch Block Message, descriptor uses different Message Type encoding																																			
	3:0	Message Type																																				
		Format: Enumeration																																				
		Specifies type of message																																				
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>MT0R_B [Default]</td> <td>Block Read message</td> </tr> <tr> <td>01h</td> <td>MT0R_AB</td> <td>Aligned Block Read message</td> </tr> <tr> <td>02h</td> <td>MT0R_OWDB</td> <td>Oword Dual Block Read message</td> </tr> <tr> <td>03h</td> <td>MT0R_DWS</td> <td>Dword Scattered Read message</td> </tr> <tr> <td>04h</td> <td>MT0R_BS</td> <td>Byte Scattered Read message</td> </tr> <tr> <td>07h</td> <td>MT0_MEMFENCE</td> <td>Memory Fence message</td> </tr> <tr> <td>08h</td> <td>MT0W_B</td> <td>Block Write message</td> </tr> <tr> <td>0Ah</td> <td>MT0W_OWDB</td> <td>Oword Dual Block Write message</td> </tr> <tr> <td>0Bh</td> <td>MT0W_DWS</td> <td>Dword Scattered Write message</td> </tr> <tr> <td>0Ch</td> <td>MT0W_BS</td> <td>Byte Scattered Write message</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Value	Name	Description	00h	MT0R_B [Default]	Block Read message	01h	MT0R_AB	Aligned Block Read message	02h	MT0R_OWDB	Oword Dual Block Read message	03h	MT0R_DWS	Dword Scattered Read message	04h	MT0R_BS	Byte Scattered Read message	07h	MT0_MEMFENCE	Memory Fence message	08h	MT0W_B	Block Write message	0Ah	MT0W_OWDB	Oword Dual Block Write message	0Bh	MT0W_DWS	Dword Scattered Write message	0Ch	MT0W_BS	Byte Scattered Write message	Others	Reserved	Ignored
		Value	Name	Description																																		
		00h	MT0R_B [Default]	Block Read message																																		
01h		MT0R_AB	Aligned Block Read message																																			
02h		MT0R_OWDB	Oword Dual Block Read message																																			
03h		MT0R_DWS	Dword Scattered Read message																																			
04h		MT0R_BS	Byte Scattered Read message																																			
07h	MT0_MEMFENCE	Memory Fence message																																				
08h	MT0W_B	Block Write message																																				
0Ah	MT0W_OWDB	Oword Dual Block Write message																																				
0Bh	MT0W_DWS	Dword Scattered Write message																																				
0Ch	MT0W_BS	Byte Scattered Write message																																				
Others	Reserved	Ignored																																				

Data Port 1 Message Types

MT_DP1 - Data Port 1 Message Types				
Source:	DataPort 1			
Size (in bits):	5			
Default Value:	0x00000000			
<p>Lists all the Message Types in a Data Port 1 Message Descriptor [18:14]. Most surface and atomic operations, both typed and untyped, are encoded on Data Port 1. The Message Header is optional for many (but not all) of these operations. Most A64 Stateless operations are also encoded on Data Port 1. The Message Header is forbidden for all A64 messages on Data Port 1.</p>				
DWord	Bit	Description		
0	4:0	Message Type		
		Format: Enumeration		
		Specifies type of message		
		Value	Name	Description
		00h	MT1R_T [Default]	Transpose Read message
		01h	MT1R_US	Untyped Surface Read message
		02h	MT1A_UI	Untyped Atomic Integer Operation message
		03h	MT1A_UI4x2	Untyped Atomic Integer Operation SIMD4x2 message
		04h	MT1R_MB	Media Block Read message
		05h	MT1R_TS	Typed Surface Read message
		06h	MT1A_TA	Typed Atomic Integer Operation message
		07h	MT1A_TA4x2	Typed Atomic Integer Operation SIMD4x2 message
		08h	Reserved	Ignored
		09h	MT1W_US	Untyped Surface Write message
		0Ah	MT1W_MB	Media Block Write message
		0Bh	MT1A_TC	Typed Atomic Counter Operation message
		0Ch	MT1A_TC4x2	Typed Atomic Counter Operation SIMD4x2 message
		0Dh	MT1W_TS	Typed Surface Write message
		0Eh	Reserved	Ignored
		10h	MT1R_A64_SB	A64 Scattered Read message
11h	MT1R_A64_US	A64 Untyped Surface Read message		
12h	MT1A_A64_UI	A64 Untyped Atomic Integer Operation message		
13h	MT1A_A64_UI4x2	A64 Untyped Atomic Integer Operation SIMD4x2 message		
14h	MT1R_A64_B	A64 Block Read message		
15h	MT1W_A64_B	A64 Block Write message		
18h	Reserved	Ignored		
19h	MT1W_A64_US	A64 Untyped Surface Write message		



MT_DP1 - Data Port 1 Message Types

		1Ah	MT1W_A64_SB	A64 Scattered Write message
		1Bh	MT1A_UF	Untyped Atomic Float Operation message
		1Ch	MT1A_UF4x2	Untyped Atomic Float Operation SIMD4x2 message
		1Dh	MT1A_A64_UF	A64 Untyped Atomic Float Operation message
		1Eh	MT1A_A64_UF4x2	A64 Untyped Atomic Float Operation SIMD4x2 message
		Others	Reserved	Ignored

Data Port 2 Extended Message Descriptor

DP2_EXTDESC - Data Port 2 Extended Message Descriptor				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:16	<p>Sideband Offset</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>Specifies the 16-bit offset from the Sideband added to all the offsets in the Address Payload for DP2 messages.</p>	Format:	U16
	Format:	U16		
	15:11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ
Format:	MBZ			
10:0	<p>Execution Unit Extended Message Descriptor Definition</p> <table border="1"> <tr> <td>Format:</td> <td>Execution_Unit_Extended_Message_Descriptor</td> </tr> </table> <p>EU uses this information as part of the SEND instruction.</p>	Format:	Execution_Unit_Extended_Message_Descriptor	
Format:	Execution_Unit_Extended_Message_Descriptor			

Data Port 2 Message Types

MT_DP2 - Data Port 2 Message Types				
Source:	DataPort 2			
Size (in bits):	5			
Default Value:	0x00000002			
<p>Lists all the Message Types in a Data Port 2 Message Descriptor [18:14]. Scaled operations are on Data Port 2. They provide a pitch-scaled data address calculation for SLM Stateless address models. The Message Header is forbidden for SLM operations.</p>				
DWord	Bit	Description		
0	4:1	Message Type		
		Format: Enumeration		
		Specifies type of message		
		Value	Name	Description
		01h	MT2R_US [Default]	Untyped Surface Read message
		02h	MT2R_A64_SB	A64 Scattered Read message
		03h	MT2R_A64_US	A64 Untyped Surface Read message
		04h	MT2R_BS	Byte Scattered Read message
		09h	MT2W_US	Untyped Surface Write message
		0Ah	MT2W_A64_US	A64 Untyped Surface Write message
	0Bh	MT2W_A64_SB	A64 Scattered Write message	
	0Ch	MT2W_BS	Byte Scattered Write message	
	Others	Reserved	Ignored	
0	0	Reserved		
		Format: MBZ Ignored		

Data Port Bindless Surface Extended Message Descriptor

DP_EXTDESC_BTI252 - Data Port Bindless Surface Extended Message Descriptor				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:12	<p>Bindless Surface Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>BindlessSurfaceOffset[25:6]</td> </tr> </table> <p>Specifies the bindless surface offset if the Binding Table Index is set to 252. Ignored otherwise. The bindless surface offset is added to the Bindless Surface Base Address as bits 25:6 of the byte-based address. The resulting address is the location of SURFACE_STATE for this message.</p>	Format:	BindlessSurfaceOffset[25:6]
	Format:	BindlessSurfaceOffset[25:6]		
	11	Reserved		
10:0	<p>Execution Unit Extended Message Descriptor Definition</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Execution_Unit_Extended_Message_Descriptor</td> </tr> </table> <p>EU uses this information as part of the SEND instruction.</p>	Format:	Execution_Unit_Extended_Message_Descriptor	
Format:	Execution_Unit_Extended_Message_Descriptor			

Data Size Message Descriptor Control Field

MDC_DS - Data Size Message Descriptor Control Field																					
Source:	BSpec																				
Size (in bits):	2																				
Default Value:	0x00000000																				
DWord	Bit	Description																			
0	1:0	<p>Data Size</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> <tr> <td colspan="2">Specifies the number of Bytes to be read or written</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">B</td> <td style="text-align: center;">1 Byte</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">W</td> <td style="text-align: center;">2 Bytes</td> </tr> <tr> <td style="text-align: center;">02h</td> <td style="text-align: center;">DW</td> <td style="text-align: center;">4 Bytes</td> </tr> <tr> <td style="text-align: center;">03h</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Reserved</td> </tr> </table>	Format:	Enumeration	Specifies the number of Bytes to be read or written		Value	Name	Description	00h	B	1 Byte	01h	W	2 Bytes	02h	DW	4 Bytes	03h	Reserved	Reserved
Format:	Enumeration																				
Specifies the number of Bytes to be read or written																					
Value	Name	Description																			
00h	B	1 Byte																			
01h	W	2 Bytes																			
02h	DW	4 Bytes																			
03h	Reserved	Reserved																			

Depth Clear Value Format

STRUCTURE_TEMPLATE - Depth Clear Value Format		
Source:	BSpec	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:0	Address1
		Format: IEEE_FP
		Format: UNORM24
		Format: UNORM16
When this field contains 24-bit UNORM, the upper 8-bits are reserved (0's) When this field contains 16-bit UNORM the upper 16-bits are reserved (0's)		

Display Engine Render Response Message Definition

Display Engine Render Response Message Definition		
Source:	BSpec	
Size (in bits):	32	
Default Value:	0x00000000	
The Display Engine Render Response Registers use bit definitions from this table.		
Programming Notes		
<p>Some events can be sent to CS (Render Command Streamer) or BCS (Blitter Command Streamer). For render response messages sending flip done or scanline events, the destination, CS or BCS, is selected depending on the initiator of the flip or the load scanline command. For render response messages sending vertical blank events, the destinations, CS or BCS, or both CS and BCS, is selected depending on the DE_RR_DEST setting. Command Streamer Plane number to the Display Plane name mapping is available in the Display Plane Capability and Interoperability section.</p> <p>The STEREO3D_EVENT_MASK selects between left eye and right eye reporting of vertical blank and scanline events in stereo 3D modes.</p>		
DWord	Bit	Description
0	31	Reserved
	30	Reserved
	29	Reserved
	28	Reserved
	27:23	Reserved
	22	Reserved
	21	Pipe_C_Start_of_Vertical_Blank_Event This event is reported on the start of the vertical blank of the transcoder attached to Pipe C.
	20	Plane_2_C_Flip_Done_Event This event is reported on the completion of a flip for Plane 2 C.
	19	Reserved
	18	Plane_4_B_Flip_Done_Event This event is reported on the completion of a flip for Plane 4 B. Not all projects have a plane 4 B.
	17	Plane_4_A_Flip_Done_Event This event is reported on the completion of a flip for Plane 4 A. Not all projects have a plane 4 A.
	16	Plane_3_C_Flip_Done_Event This event is reported on the completion of a flip for Plane 3 C.
15	Plane_1_C_Flip_Done_Event This event is reported on the completion of a flip for Plane 1 C.	

Display Engine Render Response Message Definition

14	Pipe_C_Scanline_Event This event is reported on the start of the selected scan line for the transcoder attached to Pipe C.
13	Reserved
12	Reserved
11	Pipe_B_Start_of_Vertical_Blank_Event This event is reported on the start of the vertical blank of the transcoder attached to Pipe B.
10	Plane_2_B_Flip_Done_Event This event is reported on the completion of a flip for Plane 2 B.
9	Plane_1_B_Flip_Done_Event This event is reported on the completion of a flip for Plane 1 B.
8	Pipe_B_Scanline_Event This event is reported on the start of the selected scan line for the transcoder attached to Pipe B.
7	Plane_3_B_Flip_Done_Event This event is reported on the completion of a flip for Plane 3 B.
6	Plane_3_A_Flip_Done_Event This event is reported on the completion of a flip for Plane 3 A.
5	Reserved
4	Reserved
3	Pipe_A_Start_of_Vertical_Blank_Event This event is reported on the start of the vertical blank of the transcoder attached to Pipe A.
2	Plane_2_A_Flip_Done_Event This event is reported on the completion of a flip for Plane 2 A.
1	Plane_1_A_Flip_Done_Event This event is reported on the completion of a flip for Plane 1 A.
0	Pipe_A_Scanline_Event This event is reported on the start of the selected scan line for the transcoder attached to Pipe A.

DstRegNum

DstRegNum											
Source:	Eulsa										
Size (in bits):	8										
Default Value:	0x00000000										
Description											
<p>Register Number The register number for the operand. For a GRF register, is the part of a register address that aligns to a 256-bit (32-byte) boundary. For an ARF register, this field is encoded such that MSBs identify the architecture register type and LSBs provide the register number. An ARF register can only be dst or src0. Any src1 or src2 operands cannot be ARF registers. RegNum and SubRegNum together provide the byte-aligned address for the origin of a register region. RegNum provides bits 12:5 of that address. For one-source and two-source instructions, SubregNum provides bits 4:0. For three-source instructions, the address must be DWord-aligned; SubRegNum provides bits 4:2 of the address and bits 1:0 are zero. This field is present for the direct addressing mode and not present for indirect addressing. This field applies to both source and destination operands.</p>											
DWord	Bit	Description									
0	7:0	<p>Destination Register Number</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-127</td> <td>If {Dst/Src0/Src1/Src2}.RegFile==GRF</td> <td></td> </tr> <tr> <td>0-0ffh</td> <td>If {Dst/Src0/Src1/Src2}.RegFile==ARF</td> <td>This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.</td> </tr> </tbody> </table>	Value	Name	Description	0-127	If {Dst/Src0/Src1/Src2}.RegFile==GRF		0-0ffh	If {Dst/Src0/Src1/Src2}.RegFile==ARF	This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.
Value	Name	Description									
0-127	If {Dst/Src0/Src1/Src2}.RegFile==GRF										
0-0ffh	If {Dst/Src0/Src1/Src2}.RegFile==ARF	This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.									

DstSubRegNum

DstSubRegNum											
Source:	Eulsa										
Size (in bits):	5										
Default Value:	0x00000000										
Description											
<p>Subregister Number The subregister number for the operand. For a GRF register, is the byte address within a 256-bit (32-byte) register. For an ARF register, determines the sub-register number according to the specified encoding for the given architecture register. RegNum and SubRegNum together provide the byte-aligned address for the origin of a GRF register region. RegNum provides bits 12:5 of that address. For one-source and two-source instructions, SubregNum provides bits 4:0. For three-source instructions, the address must be DWord-aligned; SubRegNum provides bits 4:2 of the address and bits 1:0 are zero.</p>											
Programming Notes											
<p>Note: The recommended instruction syntax uses subregister numbers within the GRF in units of actual data element size, corresponding to the data type used. For example for the F (Float) type, the assembler syntax uses subregister numbers 0 to 7, corresponding to subregister byte addresses of 0 to 28 in steps of 4, the element size.</p>											
DWord	Bit	Description									
0	4:0	<p>Destination Sub Register Number</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-31</td> <td>If {Dst/Src0/Src1/Src2}.RegFile==GRF</td> <td></td> </tr> <tr> <td>0-Offh</td> <td>If {Dst/Src0/Src1/Src2}.RegFile==ARF</td> <td>This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.</td> </tr> </tbody> </table>	Value	Name	Description	0-31	If {Dst/Src0/Src1/Src2}.RegFile==GRF		0-Offh	If {Dst/Src0/Src1/Src2}.RegFile==ARF	This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.
Value	Name	Description									
0-31	If {Dst/Src0/Src1/Src2}.RegFile==GRF										
0-Offh	If {Dst/Src0/Src1/Src2}.RegFile==ARF	This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.									

Dword Data Payload Register

MDCR_DW - Dword Data Payload Register				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0	31:0	Dword0 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the slot 0 data in this payload register	Format:	U32
Format:	U32			
0.1	31:0	Dword1 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the slot 1 data in this payload register	Format:	U32
Format:	U32			
0.2	31:0	Dword2 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the slot 2 data in this payload register	Format:	U32
Format:	U32			
0.3	31:0	Dword3 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the slot 3 data in this payload register	Format:	U32
Format:	U32			
0.4	31:0	Dword4 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the slot 4 data in this payload register	Format:	U32
Format:	U32			
0.5	31:0	Dword5 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the slot 5 data in this payload register	Format:	U32
Format:	U32			
0.6	31:0	Dword6 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the slot 6 data in this payload register	Format:	U32
Format:	U32			
0.7	31:0	Dword7 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the slot 7 data in this payload register	Format:	U32
Format:	U32			

Dword SIMD4x2 Atomic CMPWR Message Data Payload

MDP_AOP4X2_DW2 - Dword SIMD4x2 Atomic CMPWR Message Data Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	Src0 Slot0
		Format: U32 S31 F32 Specifies the Slot 0 Source 0 data
1	31:0	Src1 Slot0
		Format: U32 S31 F32 Specifies the Slot 0 Source 1 data
2-3	63:0	Reserved
		Format: Ignore Ignored
4	31:0	Src0 Slot1
		Format: U32 S31 F32 Specifies the Slot 1 Source 0 data
5	31:0	Src1 Slot1
		Format: U32 S31 F32 Specifies the Slot 1 Source 1 data
6-7	63:0	Reserved
		Format: Ignore Ignored

Dword SIMD4x2 Atomic Operation Message Data Payload

MDP_AOP4X2_DW1 - Dword SIMD4x2 Atomic Operation Message Data Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	<p>Dword0</p> <table border="1"> <tr> <td>Format:</td> <td>U32 S31 F32</td> </tr> </table> <p>Specifies the Slot 0 Source or Return data</p>	Format:	U32 S31 F32
Format:	U32 S31 F32			
1-3	95:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
4	31:0	<p>Dword1</p> <table border="1"> <tr> <td>Format:</td> <td>U32 S31 F32</td> </tr> </table> <p>Specifies the Slot 1 Source or Return data</p>	Format:	U32 S31 F32
Format:	U32 S31 F32			
5-7	95:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

Dword SIMD4x2 Data Payload

MDP_DW_SIMD4X2 - Dword SIMD4x2 Data Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	Red Slot0 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the Slot 0 red channel data	Format:	U32
Format:	U32			
1	31:0	Green Slot0 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the Slot 0 green channel data	Format:	U32
Format:	U32			
2	31:0	Blue Slot0 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the Slot 0 blue channel data	Format:	U32
Format:	U32			
3	31:0	Alpha Slot0 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the Slot 0 alpha channel data	Format:	U32
Format:	U32			
4	31:0	Red Slot1 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the Slot 1 red channel data	Format:	U32
Format:	U32			
5	31:0	Green Slot1 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the Slot 1 green channel data	Format:	U32
Format:	U32			
6	31:0	Blue Slot1 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the Slot 1 blue channel data	Format:	U32
Format:	U32			
7	31:0	Alpha Slot1 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the Slot 1 alpha channel data	Format:	U32
Format:	U32			

Dword SIMD8 Atomic Operation CMPWR Message Data Payload

MDP_AOP8_DW2 - Dword SIMD8 Atomic Operation CMPWR Message Data Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Src0
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDCR_DW</td> </tr> </table> <p>Specifies the Slot [7:0] Source 0 data</p>
Format:	MDCR_DW	
1.0-1.7	255:0	Src1
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDCR_DW</td> </tr> </table> <p>Specifies the Slot [7:0] Source 1 data</p>
Format:	MDCR_DW	

Dword SIMD8 Data Payload

MDP_DW_SIMD8 - Dword SIMD8 Data Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	Data[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDCR_DW</td> </tr> </table> Specifies the Slot [7:0] data	Format:	MDCR_DW
Format:	MDCR_DW			

Dword SIMD16 Data Payload

MDP_DW_SIMD16 - Dword SIMD16 Data Payload				
Source:	BSpec			
Size (in bits):	512			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	Data[7:0] <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDCR_DW</td> </tr> </table> Specifies the Slot [7:0] data	Format:	MDCR_DW
Format:	MDCR_DW			
1.0-1.7	255:0	Data[15:8] <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDCR_DW</td> </tr> </table> Specifies the Slot [15:8] data	Format:	MDCR_DW
Format:	MDCR_DW			

DX9_CONSTANTB_ENTRY

DX9_CONSTANTB_ENTRY				
Source:	RenderCS			
Size (in bits):	32			
Default Value:	0x00000000			
This structure is the payload of the 3DSTATE_DX9_CONSTANTB_* commands. Each entry provides the values for the one boolean constant being updated.				
DWord	Bit	Description		
0	31:0	<p>Component</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>The boolean value to be stored.</p>	Format:	U32
Format:	U32			

DX9_CONSTANTF_ENTRY

DX9_CONSTANTF_ENTRY				
Source:	RenderCS			
Size (in bits):	128			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000			
<p>This structure is the payload of the 3DSTATE_DX9_CONSTANTF_* commands. Each entry provides the values for the four components of one float constant being updated.</p>				
DWord	Bit	Description		
0	127:96	<p>Component 3</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>The 4th component of the nth float to be stored.</p>	Format:	IEEE_Float
	Format:	IEEE_Float		
	95:64	<p>Component 2</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>The 3rd component of the nth float to be stored.</p>	Format:	IEEE_Float
	Format:	IEEE_Float		
63:32	<p>Component 1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>The 2nd component of the nth float to be stored.</p>	Format:	IEEE_Float	
Format:	IEEE_Float			
31:0	<p>Component 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_Float</td> </tr> </table> <p>The 1st component of the nth float to be stored.</p>	Format:	IEEE_Float	
Format:	IEEE_Float			

DX9_CONSTANTI_ENTRY

DX9_CONSTANTI_ENTRY				
Source:	RenderCS			
Size (in bits):	128			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000			
<p>This structure is the payload of the 3DSTATE_DX9_CONSTANTI_* commands. Each entry provides the values for the four components of one integer constant being updated.</p>				
DWord	Bit	Description		
0	31:0	<p>Component 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> <p>The 1st component of the nth float to be stored.</p>	Format:	U32
Format:	U32			
1	31:0	<p>Component 1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> <p>The 2nd component of the nth float to be stored.</p>	Format:	U32
Format:	U32			
2	31:0	<p>Component 2</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> <p>The 3rd component of the nth float to be stored.</p>	Format:	U32
Format:	U32			
3	31:0	<p>Component 3</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> <p>The 4th component of the nth float to be stored.</p>	Format:	U32
Format:	U32			

Encoder Control State Parameters0

Encoder Control State Parameters0												
Source:	BSpec											
Size (in bits):	320											
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000											
DWord	Bit	Description										
0	31:30	Reserved Format: MBZ										
	29	Reserved										
	28:27	AVC Encoder Chroma Sub-sample type AVC YUV chroma compression mode. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>YUV 4:2:0</td> </tr> <tr> <td>01b</td> <td>RGBA 4:4:4:4</td> </tr> <tr> <td>10b</td> <td>YUV 4:4:4</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	00b	YUV 4:2:0	01b	RGBA 4:4:4:4	10b	YUV 4:4:4	11b	Reserved
	Value	Name										
	00b	YUV 4:2:0										
	01b	RGBA 4:4:4:4										
	10b	YUV 4:4:4										
	11b	Reserved										
	26:5	Reserved Format: MBZ										
	4	Reserved										
3	Reserved Format: MBZ											
2	Conditional Replenishment Enable Format: U1 If the distortion for a MB is below a fixed threshold, it is coded as Pskip (No Coded Coeff). If above threshold, MB is coded as Intra type. <p style="text-align: center;">Programming Notes</p> This field must be set once per session. If this field is off, all frames are encoded as IDR frames, and frame number as 0. GOP Size parameter must be set to 1. This field should be turned off when IPCM is on.											
1	Reserved Format: MBZ											
0	Reserved											
1	31:17	Reserved Format: MBZ										
	16	GOP-level indirect user-defined NAL data packet - enable GOP-level user defined NAL packet is inserted during the first frame of a GOP if this field is										

Encoder Control State Parameters0

		enabled.	
		Value	Name
		Description	
		0	Disable
		1	Enable
		Programming Notes	
		<p>Recommended value is 0. Header contains several NAL units. AVC spec needs AUD NAL to be the first NAL for a frame. If GOP header and Frame headers are enabled and both have AUD NAL it would lead to incorrect bit stream.</p> <p>Multiple NAL data packets can be programmed within the GOP-level indirect surface. A linked list data structure is allocated multiple elements (NAL packets) in contiguous memory. Each element consists of link descriptor fields and datum. The first four bytes of each element contain the link descriptor fields: Byte 0: length_in_bytes[7:0] -- LSB length of data element/NAL packet. Byte 1: length_in_bytes[15:8] -- MSB length of data element/NAL packet. Byte 2: <{7'b000_0000, last_NAL} -- Bit 0 indicates the current element is the last element of the link list. Byte 3: Reserved -- Reserved- MBZ Datum starts at Byte 4 of each element: Byte 4: NAL_START Byte 5: ... Byte N: NAL_END Case 1: last_NAL = 1 all the bytes through ByteN would be sent to BSP. Case 2: last_NAL = 0 and ByteN is CL aligned (N mod 64 = 0). Next NAL descriptor starts at N + 1 Byte. Case 3: last_NAL = 0 and ByteN is not CL aligned zero bytes stuffed till next NAL descriptor (63 - (N mod 64) bytes of zeros). Note: empty/zero datum is not allowed</p>	
	15:0	Reserved	
2	31:17	Reserved	
		Format:	MBZ

Encoder Control State Parameters0				
16	Frame-level indirect user-defined NAL data packet - enable Frame-level user defined NAL packet is inserted during the first frame of a frame if this field is enabled.			
	Value	Name		
	0	Disable		
	1	Enable		
Programming Notes				
The recommended value is 1. As mentioned in GOP header enable, enabling GOP header and frame header causes problem. Enabling only frame header is safe and sufficient.				
Multiple NAL data packets can be programmed within the frame-level indirect surface. A linked list data structure is allocated multiple elements (NAL packets) in contiguous memory. Each element consists of link descriptor fields and datum. The first four bytes of each element contain the link descriptor fields: Byte 0: length_in_bytes[7:0] -- LSB length of data element/NAL packet. Byte 1: length_in_bytes[15:8] -- MSB length of data element/NAL packet. Byte 2: <{7'b000_0000, last_NAL} -- Bit 0 indicates the current element is the last element of the link list. Byte 3: Reserved -- Reserved- MBZ Datum starts at Byte 4 of each element: Byte 4: NAL_START Byte 5: ... Byte N: NAL_END Note: empty/zero datum is not allowed				
15:0	Reserved.			
3	31:0	Reserved		
		Format: MBZ		
4	31:0	Reserved		
		Format: MBZ		
5	31:16	GOP size parameter Number of (P-Frames + 1) per GOP structure.		
		Value	Name	Description
		0		Infinite number of P-frames followed by an initial I-frame(IDR)
		1		GOP structure will contain one IDR frame. HME and IME inter predictions are disabled.
	2		One IDR + one P-frame per GOP	
3-FFFFh		This parameter indicates the number of frames within the GOP structure. Each GOP will have one IDR frame follows by the value of the parameter minus one P-frames.		
15:4	Reserved			
Format: MBZ				

Encoder Control State Parameters0											
	3:0	<p>log2 Maximum Frame Number Minus4</p> <p>This field specifies the value of the variable MaxFrameNum that is used in Frame_Num count for every frame and must match the definition of log2_max_frame_num_minus field in SPS NAL packet.</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 style="text-align: center;">0-12</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>The same value should be set for log2_max_pic_order_cnt_lsb_minus4 field in SPS NAL packet.</p>	Value	Name	0-12						
	Value	Name									
	0-12										
	6	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										
7	31	<p>Transform 8x8 Flag</p> <p>This field indicates that 8x8 transform can be used within the frame.</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 style="text-align: center;">0</td> <td>4x4 Integer Transform</td> <td>The MB must be set to 4x4 transform.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>8x8 Integer Transform</td> <td>The MB <u>could</u> be coded with 8x8 transform.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>When this field is set to 1, The transform_size_8x8_flag syntax element, if present in the output bit stream, is the same as this field; However, whether transform_size_8x8_flag is present or not in the output bit stream depends on several other conditions.</p> <p>Hardware set MB level transform8x8 flag to 1 for two conditions:</p> <ul style="list-style-type: none"> • It might be 1 if IntraMbFlag = INTRA and IntraMbMode = INTRA_8x8 • It must be 1 if IntraMbFlag = INTER and there is no sub partition size less than 8x8 <p>Otherwise, this field must be set to 0.</p> <p>For Intra MB if transform8x8 = 0, hardware will always honor it, thus intra8x8 will be the winner.</p> <p>If transform8x8 = 0, Intra16x16 prediction Enable or Intra4x4 prediction Enable MUST be true.</p>	Value	Name	Description	0	4x4 Integer Transform	The MB must be set to 4x4 transform.	1	8x8 Integer Transform	The MB <u>could</u> be coded with 8x8 transform.
Value	Name	Description									
0	4x4 Integer Transform	The MB must be set to 4x4 transform.									
1	8x8 Integer Transform	The MB <u>could</u> be coded with 8x8 transform.									
	30:29	<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										

Encoder Control State Parameters0			
28:24	<p>Second Chroma QP Offset, Chroma_qp_offset [9:5]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S4</td> </tr> </table> <p>Range: -12 to +12 According to AVC Spec.</p> <p>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.</p> <ul style="list-style-type: none"> Chroma_qp_offset [9:5] - second_chroma_qp_offset_bits <p style="text-align: center;">Programming Notes</p> <p><u>To ensure that the MB size doesn't exceed 3200 bits</u>, Cr/Cb QP cannot go below 10. The Value of MinQp for Luma and Chroma Offset is programmed in such a way to ensure this. E.g. If chroma offset = -5, MinQp should be ≥ 15. This would ensure that the Final Chroma QP $\geq (-5+15 = 10)$.</p>	Format:	S4
Format:	S4		
23:21	<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		
20:16	<p>Chroma QP Offset, Chroma_qp_offset[4:0]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S4</td> </tr> </table> <p>Range: -12 to +12 According to AVC Spec.</p> <p>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.</p> <ul style="list-style-type: none"> Chroma_qp_offset [4:0] - chroma_qp_offset_bits (from the current active PPS) <p style="text-align: center;">Programming Notes</p> <p><u>To ensure that the MB size doesn't exceed 3200 bits</u>, Cr/Cb QP cannot go below 10. The Value of MinQp for Luma and Chroma Offset is programmed in such a way to ensure this. E.g. If chroma offset = -5, MinQp should be ≥ 15. This would ensure that the Final Chroma QP $\geq (-5+15 = 10)$.</p>	Format:	S4
Format:	S4		
15:8	<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>Round Inter Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p style="text-align: center;">Programming Notes</p> <p>Recommended driver setting is 0. When rounding Inter is disabled, a value of 2/8 is used for rounding inter coefficients.</p>	Format:	Enable
Format:	Enable		

Encoder Control State Parameters0		
	6:4	Rounding Inter (N) Format: RoundingPrecisionTable_3_Bits Programming Notes Hardware default this field to 2 if "Round Inter Enable" is disable.
	3	Round Intra Enable Format: Enable Programming Notes Recommended driver setting is 0. When rounding Intra is disabled, a value of 4/8 is used for rounding intra coefficients.
	2:0	Rounding Intra (N) Format: RoundingPrecisionTable_3_Bits Programming Notes Hardware default this field to 4 if "Round Intra Enable" is disable.
8	31:0	Reserved Format: MBZ
9	31:19	Reserved Format: MBZ

Encoder Control State Parameters0

18:16 Slice Pattern Per MB Row

The field should be set according to the following table and the MB row size. In cases where row cannot be divided evenly, round up to the nearest MB to achieve the indicated number of slices per row.

For purposes of this clause, the rate of macroblocks/second is the rate that applies to the video being transmitted, according to the definition in the AVC standard, That is:
 the macroblock rate = ceiling(frame width, 16)/16 * ceiling(frame height, 16)/16 * (frames/second).

Slice size Must be larger than 4MB.

Macroblocks/second Rate	Slice Pattern
rate < 250,000	1 row = 1 slice
250,000 <= rate < 500,000	1 row = 2 equal slices
500,000 <= rate < 1,000,000	1 row = 4 equal slices
1,000,000 <= rate < 2,100,000	1 row = 8 equal slices
2,100,000 <= rate	1 row = 16 equal slices

Value	Name	Description
0		1 slice per MB row
1		2 slice per MB row
2		4 slice per MB row
3		8 slice per MB row
4		16 slice per MB row

15:0 Reserved

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

Encoder Statistics Format

Encoder Statistics Format						
Source:	VideoEnhancementCS					
Size (in bits):	128					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000					
Description						
<p>The per block data is intended for use by the video encoder and consists of 16 bytes of Denoise block data and FMD variances. Much of the data is encoded as an 8-bit mantissa with the leading 1 removed and a 4-bit shift. To recover the original 17-bit integer this code can be used: If (exp != 0) Number = ((0x100 Mantissa) « exp) » 7; else Number = mantissa;</p> <p>The values for STAD, SHCM and SVCM for each 4x4 are shifted down by 2 bits to make 14-bit values before being summed for the 16x4 block to make a 16-bit value. The result is then converted into the mantissa/exp format.</p>						
DWord	Bit	Description				
0	31:24	Tearing_Count 1 (FMD Variance[8])				
		Format: U8				
		Number of pixels that have (diff_cTcB > diff_cTcT + diff_cBcB)				
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>DI is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0
Value	Name	Description				
0		DI is Disabled				
23:16		Tearing_Count 2				
		Format: U8				
		<p>If the frame is Deinterlaced with Top First in the DN/DI state then this is (FMD Variance[9]) = Number of pixels that have (diff_cTpB > diff_cTcT + diff_pBpB)</p> <p>If the frame is bottom first then this is (FMD Variance[10]) = Number of pixels that have (diff_cBpT > diff_pTpT + diff_cBcB)</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>DI is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0
Value	Name	Description				
0		DI is Disabled				
15:8		Motion_Count (FMD Variance[7])				
		Format: U8				
		Number of pixels that are moving (different above a threshold)				
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>DI is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0
Value	Name	Description				
0		DI is Disabled				
7:0		Reserved				
		Format: MBZ				

Encoder Statistics Format												
1	31:28	<p>sSTAD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> <tr> <td colspan="2">Shift for the Sum in time of absolute differences for 16x4.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>Temporal Denoise Filtering is Disabled.</td> </tr> </table>	Format:	U4	Shift for the Sum in time of absolute differences for 16x4.		Value	Name	Description	0		Temporal Denoise Filtering is Disabled.
	Format:	U4										
	Shift for the Sum in time of absolute differences for 16x4.											
	Value	Name	Description									
	0		Temporal Denoise Filtering is Disabled.									
	27:24	<p>sSHCM</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> <tr> <td colspan="2">Shift for the Sum horizontal of absolute differences.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>DN is Disabled</td> </tr> </table>	Format:	U4	Shift for the Sum horizontal of absolute differences.		Value	Name	Description	0		DN is Disabled
	Format:	U4										
Shift for the Sum horizontal of absolute differences.												
Value	Name	Description										
0		DN is Disabled										
23:20	<p>sSVCM</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> <tr> <td colspan="2">Shift for the Sum vertically of absolute differences.</td> </tr> </table>	Format:	U4	Shift for the Sum vertically of absolute differences.								
Format:	U4											
Shift for the Sum vertically of absolute differences.												
19:16	<p>sDiff_cTpT</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> <tr> <td colspan="2">Shift for the sum of differences in top fields of current and previous frame.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>DI is Disabled</td> </tr> </table>	Format:	U4	Shift for the sum of differences in top fields of current and previous frame.		Value	Name	Description	0		DI is Disabled	
Format:	U4											
Shift for the sum of differences in top fields of current and previous frame.												
Value	Name	Description										
0		DI is Disabled										
15:12	<p>sDiff_cBpB</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> <tr> <td colspan="2">Shift for the sum of differences in bottom field of current and previous frame.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>DI is Disabled</td> </tr> </table>	Format:	U4	Shift for the sum of differences in bottom field of current and previous frame.		Value	Name	Description	0		DI is Disabled	
Format:	U4											
Shift for the sum of differences in bottom field of current and previous frame.												
Value	Name	Description										
0		DI is Disabled										
11:8	<p>sDiff_cTcB</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> <tr> <td colspan="2">Shift for the sum of differences between top and bottom field in current frame.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>DI is Disabled</td> </tr> </table>	Format:	U4	Shift for the sum of differences between top and bottom field in current frame.		Value	Name	Description	0		DI is Disabled	
Format:	U4											
Shift for the sum of differences between top and bottom field in current frame.												
Value	Name	Description										
0		DI is Disabled										
7:4	<p>sDiff_cTpB</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> <tr> <td colspan="2">Shift for the sum of differences between current top and previous bottom.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>DI is Disabled</td> </tr> </table>	Format:	U4	Shift for the sum of differences between current top and previous bottom.		Value	Name	Description	0		DI is Disabled	
Format:	U4											
Shift for the sum of differences between current top and previous bottom.												
Value	Name	Description										
0		DI is Disabled										

Encoder Statistics Format								
2	3:0	sDiff_cBpT Format: U4 Shift for the sum of differences between current bottom and previous top. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>DI is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0		DI is Disabled
	Value	Name	Description					
	0		DI is Disabled					
	31:24	mDiff_cBpB (FMD Variance[1]) Format: U8 Mantissa of sum of differences in bottom field of current and previous frame. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>DI is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0		DI is Disabled
	Value	Name	Description					
0		DI is Disabled						
23:16	mDiff_cTcB (FMD Variance[2]) Format: U8 Mantissa of sum of differences between top and bottom field in current frame. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>DI is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0		DI is Disabled	
Value	Name	Description						
0		DI is Disabled						
15:8	mDiff_cTpB (FMD Variance[3]) Format: U8 Mantissa of sum of differences between current top and previous bottom. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>DI is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0		DI is Disabled	
Value	Name	Description						
0		DI is Disabled						
7:0	mDiff_cBpT (FMD Variance[4]) Format: U8 Mantissa of sum of differences between current bottom and previous top. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>DI is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0		DI is Disabled	
Value	Name	Description						
0		DI is Disabled						
3	31:24	mSTAD Format: U8 Mantissa of Sum in time of absolute differences for 16x4. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Temporal Denoise Filtering is disabled.</td> </tr> </tbody> </table>	Value	Name	Description	0		Temporal Denoise Filtering is disabled.
	Value	Name	Description					
0		Temporal Denoise Filtering is disabled.						
23:16	mSHCM Format: U8 Mantissa of Sum horizontal of absolute differences. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>DN is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0		DN is Disabled	
Value	Name	Description						
0		DN is Disabled						

Encoder Statistics Format						
	15:8	mSVCM				
		Format: U8				
		Mantissa of Sum vertically of absolute differences.				
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%; text-align: center;">Value</th> <th style="width: 33%; text-align: center;">Name</th> <th style="width: 33%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>DN is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0
Value	Name	Description				
0		DN is Disabled				
	7:0	mDiff_cTpT (FMD Variance[0])				
		Format: U8				
		Mantissa of sum of differences in top fields of current and previous frame.				
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%; text-align: center;">Value</th> <th style="width: 33%; text-align: center;">Name</th> <th style="width: 33%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>DI is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0
Value	Name	Description				
0		DI is Disabled				

EU_INSTRUCTION_BASIC_ONE_SRC

EU_INSTRUCTION_BASIC_ONE_SRC		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
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	

EU_INSTRUCTION_BASIC_THREE_SRC

EU_INSTRUCTION_BASIC_THREE_SRC																							
Source:	Eulsa																						
Size (in bits):	128																						
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000																						
DWord	Bit	Description																					
0..3	127	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td>MBZ</td></tr></table>		MBZ																			
		MBZ																					
	126:106	Source 2 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC																					
	105:85	Source 1 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC																					
	84:64	Source 0 Format: EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC																					
	63:56	Destination Register Number Format: DstRegNum																					
	55:53	Destination Subregister Number																					
	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																					
	48:46	Destination Data Type <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%;">Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>:f</td> <td>single precision Float (32-bit)</td> </tr> <tr> <td>001b</td> <td>:d</td> <td>signed Doubleword integer</td> </tr> <tr> <td>010b</td> <td>:ud</td> <td>Unsigned Doubleword integer</td> </tr> <tr> <td>011b</td> <td>:df</td> <td>Double precision Float (64-bit)</td> </tr> <tr> <td>100b</td> <td>:hf</td> <td>Half Float (16-bit)</td> </tr> <tr> <td>101b-111b</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	000b	:f	single precision Float (32-bit)	001b	:d	signed Doubleword integer	010b	:ud	Unsigned Doubleword integer	011b	:df	Double precision Float (64-bit)	100b	:hf	Half Float (16-bit)	101b-111b	Reserved	
	Value	Name	Description																				
000b	:f	single precision Float (32-bit)																					
001b	:d	signed Doubleword integer																					
010b	:ud	Unsigned Doubleword integer																					
011b	:df	Double precision Float (64-bit)																					
100b	:hf	Half Float (16-bit)																					
101b-111b	Reserved																						

EU_INSTRUCTION_BASIC_THREE_SRC

45:43	Source Data Type	
	Value	Name
	000b	:f
	001b	:d
	010b	:ud
	011b	:df
	100b	:hf
	101b-111b	Reserved
42:41	Source 2 Modifier	
	Exists If:	(Property[Source Modifier] == 'true')
	Format:	SrcMod
40:39	Source 1 Modifier	
	Exists If:	(Property[Source Modifier] == 'true')
	Format:	SrcMod
42:37	Reserved	
	Exists If:	(Property[Source Modifier] == 'false')
	Format:	MBZ
38:37	Source 0 Modifier	
	Exists If:	(Property[Source Modifier] == 'true')
	Format:	SrcMod
36	Source 1 Type	
	Format:	U1
	Only used if Source Data Type is :f or :hf, else Source 1 Data Type matches Source 0 type and this bit is ignored.	
	Value	Name
	0b	:f
	1b	:hf
		Description
		single precision Float (32-bit)
		Half Float (16-bit)
35	Source 2 Type	
	Format:	U1
	Only used if Source Data Type is :f or :hf, else Source 2 Data Type matches Source 0 type and this bit is ignored.	
	Value	Name
	0b	:f
	1b	:hf
		Description
		single precision Float (32-bit)
		Half Float (16-bit)

EU_INSTRUCTION_BASIC_THREE_SRC											
34	<p>MaskCtrl (formerly WECtrl/Write Enable Control). This flag disables the normal write enables; it should normally be 0.</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 style="text-align: center;">Normal</td> <td>Use the normal write enables in Dst.ChanEn (normal setting).</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">NoMask</td> <td>Write all channels except those disabled by predication or by other masks besides the write enables.</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>MaskCtrl = NoMask also skips the check for PclP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.</p>		Value	Name	Description	0	Normal	Use the normal write enables in Dst.ChanEn (normal setting).	1	NoMask	Write all channels except those disabled by predication or by other masks besides the write enables.
Value	Name	Description									
0	Normal	Use the normal write enables in Dst.ChanEn (normal setting).									
1	NoMask	Write all channels except those disabled by predication or by other masks besides the write enables.									
33	<p>Flag Register Number This field contains the flag register number for instructions with a non-zero Conditional Modifier.</p>										
32	<p>Flag Subregister Number This field contains the flag subregister number for instructions with a non-zero Conditional Modifier.</p>										
31:0	<p>Header</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td style="text-align: center;">EU_INSTRUCTION_HEADER</td> </tr> </table>		Format:	EU_INSTRUCTION_HEADER							
Format:	EU_INSTRUCTION_HEADER										

EU_INSTRUCTION_BASIC_TWO_SRC

EU_INSTRUCTION_BASIC_TWO_SRC		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
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	

EU_INSTRUCTION_BRANCH_CONDITIONAL

EU_INSTRUCTION_BRANCH_CONDITIONAL		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..3	127:64	Sources
		Exists If: ([Src1.RegFile]!='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_REG	
	127:64	Sources
		Exists If: ([Src1.RegFile]='IMM')
	Format: EU_INSTRUCTION_SOURCES_REG_IMM	
	63:48	JIP
		Format: S15
	Jump Target Offset. The jump distance in number of eight-byte units if a jump is taken for the instruction.	
	47	Reserved
Format: MBZ		
46:44	Src1.SrcType	
	Format: DataType	
	This field specifies the numeric data type of the source operand src1. The bits of a source operand are interpreted as the identified numeric data type, rather than coerced into a type implied by the operator. Depending on RegFile field of the source operand, there are two different encoding for this field. If a source is a register operand, this field follows the Source Register Type Encoding. If a source is an immediate operand, this field follows the Source Immediate Type Encoding.	
	<p style="text-align: center;">Programming Notes</p> <p>Both source operands, src0 and src1, support immediate types, but only one immediate is allowed for a given instruction and it must be the last operand.</p> <p>Halfbyte integer vector (v) type can only be used in instructions in packed-word execution mode. Therefore, in a two-source instruction where src1 is of type :v, src0 must be of type :b, :ub, :w, or :uw.</p>	
43:42	Src1.RegFile	
	Format: RegFile	
41:39	Src0.SrcType	
	Format: DataType	
38:37	Src0.RegFile	
	Format: RegFile	

EU_INSTRUCTION_BRANCH_CONDITIONAL			
36:34	<p>Destination Data Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">DataType</td> </tr> </table> <p>This field specifies the numeric data type of the destination operand dst. The bits of the destination operand are interpreted as the identified numeric data type, rather than coerced into a type implied by the operator. For a send instruction, this field applies to the CurrDst ? the current destination operand.</p>	Format:	DataType
Format:	DataType		
33:32	<p>Destination Register File</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">RegFile</td> </tr> </table>	Format:	RegFile
Format:	RegFile		
31:0	<p>Header</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>EU_INSTRUCTION_HEADER</td> </tr> </table>	Format:	EU_INSTRUCTION_HEADER
Format:	EU_INSTRUCTION_HEADER		

EU_INSTRUCTION_BRANCH_ONE_SRC

EU_INSTRUCTION_BRANCH_ONE_SRC		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
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
		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

EU_INSTRUCTION_BRANCH_TWO_SRC

EU_INSTRUCTION_BRANCH_TWO_SRC		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
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

EU_INSTRUCTION_COMPACT_THREE_SRC

EU_INSTRUCTION_COMPACT_THREE_SRC				
Source:	Eulsa			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0..1	63:57	<p>Src2.RegNum[6:0]</p> <table border="1"> <tr> <td>Format:</td> <td>SrcRegNum[6:0]</td> </tr> </table> <p>Src2.RegNum[6:0]. The SourceIndex field in the compact instruction determines Src2.RegNum[7].</p> <p>Maps to 124:118</p>	Format:	SrcRegNum[6:0]
	Format:	SrcRegNum[6:0]		
	56:50	<p>Src1.RegNum[6:0]</p> <table border="1"> <tr> <td>Format:</td> <td>SrcRegNum[6:0]</td> </tr> </table> <p>Src1.RegNum[6:0]. The SourceIndex field in the compact instruction determines Src1.RegNum[7].</p> <p>Maps to 103:97</p>	Format:	SrcRegNum[6:0]
	Format:	SrcRegNum[6:0]		
	49:43	<p>Src0.RegNum[6:0]</p> <table border="1"> <tr> <td>Format:</td> <td>SrcRegNum[6:0]</td> </tr> </table> <p>Src0.RegNum[6:0]. The SourceIndex field in the compact instruction determines Src0.RegNum[7].</p> <p>Maps to 82:76</p>	Format:	SrcRegNum[6:0]
	Format:	SrcRegNum[6:0]		
	42:40	<p>Src2.SubRegNum</p> <table border="1"> <tr> <td>Format:</td> <td>SrcSubRegNum[4:2]</td> </tr> </table> <p>Maps to 117:115</p>	Format:	SrcSubRegNum[4:2]
	Format:	SrcSubRegNum[4:2]		
39:37	<p>Src1.SubRegNum</p> <table border="1"> <tr> <td>Format:</td> <td>SrcSubRegNum[4:2]</td> </tr> </table> <p>Maps to 96:94</p>	Format:	SrcSubRegNum[4:2]	
Format:	SrcSubRegNum[4:2]			
36:34	<p>Src0.SubRegNum</p> <table border="1"> <tr> <td>Format:</td> <td>SrcSubRegNum[4:2]</td> </tr> </table> <p>Maps to 75:73</p>	Format:	SrcSubRegNum[4:2]	
Format:	SrcSubRegNum[4:2]			
33	<p>Src2.RepCtrl</p> <table border="1"> <tr> <td>Format:</td> <td>RepCtrl</td> </tr> </table> <p>Maps to 106</p>	Format:	RepCtrl	
Format:	RepCtrl			

EU_INSTRUCTION_COMPACT_THREE_SRC			
32	Src1.RepCtrl		
	Format:	RepCtrl	
Maps to 85			
31	Reserved		
	Exists If:	(Property[Saturation]='false')	
Format:	MBZ		
31	Saturate		
	Exists If:	(Property[Saturation]='true')	
Maps to 31			
30	Reserved		
29	Compaction Control		
	Format:	CmptCtrl	
28	Src0.RepCtrl		
	Format:	RepCtrl	
Maps to 64			
27:19	Reserved		
	Format:	MBZ	
18:12	Dst.RegNum[6:0]		
	Format:	DstRegNum[6:0]	
	Dst.RegNum[7:0] with MSB of zero and [6:0] from the compact instruction		
	Maps to 63:56 (Dst.RegNum)		
11:10	SourceIndex		
	Lookup one of four 46-bit values. That value is used (from MSB to LSB) for the Src2.RegNum[7], Src1.RegNum[7], Src0.RegNum[7], Src2.ChanSel, Src1.ChanSel, Src0.ChanSel, Dst.SubRegNum, Dst.ChanEnable, Dst.DstType, SrcType, Src2.Modifier, Src1.Modifier, and Src0.Modifier bit fields.		
	Maps to 125, 104, 83, 114:107, 93:86, 72:65, 55:49, 48:43, 42:37		
	Value	Name	Description
	0	0001110010011100100111001000001111000000000000	No Negation
1	0001110010011100100111001000001111000000000010	Negate Src0	
2	0001110010011100100111001000001111000000001000	Negate Src1	
3	0001110010011100100111001000001111000000100000	Negate Src2	

EU_INSTRUCTION_COMPACT_THREE_SRC		
9:8	ControlIndex Lookup one of four 24-bit values. That value is used (from MSB to LSB) for the MaskCtrl, FlagRegNum/FlagSubRegNum, AccWrCtrl, CondModifier, ExecSize, PredInv, PredCtrl, ThreadCtrl, QtrCtrl, NibCtrl, DepCtrl, and AccessMode bit fields. Maps to 34, 33:32, 28:8	
	Value	Name
	0	1000000001100000000000001
	1	0000000001100000000000001
	2	0000000010000000000000001
	3	0000000010000000000100001
	Description	
		(8) Q1 NoMask Align16
		(8) Q1 Align16
		(16) H1 Align16
		(16) H2 Align16
7	Reserved Format: MBZ	
6:0	Opcode	

EU_INSTRUCTION_COMPACT_TWO_SRC

EU_INSTRUCTION_COMPACT_TWO_SRC		
Source:	Eulsa	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
<p>The following table describes the EU compact instruction format. The compact instruction format for 1 or 2-source instructions is essentially identical to the compact instruction format for earlier generations, but the compact fields expand to somewhat different fields in the native instruction format, as the native instruction format changed</p>		
DWord	Bit	Description
0..1	63:56	Src1.RegNum
		Exists If: ([DataTypeIndex][Src1.RegFile]!='IMM')
		Format: SrcRegNum
	Maps to 108:101 (Src1.RegNum)	
	63:56	Src1.RegNum
		Exists If: ([DataTypeIndex][Src1.RegFile]='IMM')
	Maps to 103:96 (Imm32[7:0])	
	55:48	Src0.RegNum
		Format: SrcRegNum
	Maps to 76:69 (Src0.RegNum)	
47:40	Dst.RegNum	
	Format: DstRegNum	
Maps to 60:53 (Dst.RegNum)		
39:35	Src1 Index	
	Exists If: ([DataTypeIndex][Src1.RegFile]!='IMM')	
	Format: SrcIndex	
<p>If not an immediate operand, lookup one of 32 12-bit values that maps to bits 120:109. That value is used (from MSB to LSB) for the Src1.VertStride, various Src1 bit fields based on AccessMode (Src1.ChanSel[7:4], Src1.Width, Src1.HorzStride), Src1.AddrMode, and Src1.SrcMod bit fields</p>		
Maps to 120:109		
39:35	Src1 Index	
	Exists If: ([DataTypeIndex][Src1.RegFile]='IMM')	
	<p>If an immediate operand, there is no lookup. Determines bits 127:104 (Imm32[31:8]) as follows: map bits 39:35 directly to bits 108:104. Sign extend to fill bits 127:109. Compact format bit 39 is thus copied to all of bits 127:108 for an immediate operand.</p>	
Maps to 127:104		

EU_INSTRUCTION_COMPACT_TWO_SRC			
34:30	Src0Index		
	Format:	SrcIndex	
<p>Lookup one of 32 12-bit values. That value is used (from MSB to LSB) for the Src0.VertStride, various Src0 bit fields based on AccessMode (Src0.ChanSel[7:4], Src0.Width, Src0.HorzStride), Src0.AddrMode, and Src0.SrcMod bit fields. Note that this field spans a DWord boundary within the QWord compacted instruction.</p>			
Maps to 88:77			
29	Compaction Control		
	Format:	CmptCtrl	
28	Reserved		
	Format:	MBZ	
27:24	Reserved		
	Exists If:	(Property[Conditional Modifier]='false')	
	Format:	MBZ	
27:24	Conditional Modifier		
	Exists If:	(Property[Conditional Modifier]='true')	
	Format:	CondModifier	
23	Accumulator Write Control		
	Format:	AccWrCtrl	
22:18	SubRegIndex		
	<p>Lookup one of 32 15-bit values. That value is used (from MSB to LSB) for various fields for Src1, Src0, and Dst, including ChanEn/ChanSel, SubRegNum, and AddrImm[4] or AddrImm[4:0], depending on AddrMode and AccessMode.</p>		
	Maps to 100:96, 68:64, 52:48		
	Value	Name	Description
	0	0000000000000000	0 0 0
	1	0000000000000001	0.x 0.xx 0.xx
	2	0000000000010000	8 0 0
	3	0000000000011111	0.xyzw 0.xx 0.xx
	4	0000000001000000	16 0 0
	5	0000000100000000	0 4 0
	6	0000001000000000	0 8 0
	7	0000001100000000	0 12 0
8	0000010000000000	0 16 0	
9	0000010000100000	16 16 0	
10	0000010100000000	0 20 0	

EU_INSTRUCTION_COMPACT_TWO_SRC

		11	0010000000000000	0 0 4																														
		12	0010000000000001	0.x 0.xx 0.xy																														
		13	0010000100000001	0.x 0.xy 0.xy																														
		14	0010000100000010	0.y 0.xy 0.xy																														
		15	0010000100000011	0.xy 0.xy 0.xy																														
		16	0010000100001000	0.z 0.xy 0.xy																														
		17	0010000100001011	0.xyz 0.xy 0.xy																														
		18	0010000100010000	0.w 0.xy 0.xy																														
		19	0010000100011100	0.yzw 0.xy 0.xy																														
		20	0010000100011111	0.xyzw 0.xy 0.xy																														
		21	0010001100000000	0 12 4																														
		22	0010001111010000	0.w 0.ww 0.xy																														
		23	0100000000000000	0 0 8																														
		24	0100001100000000	0 12 8																														
		25	0110000000000000	0 0 12																														
		26	0111100100001111	0.xyz 0.xy 0.ww																														
		27	1000000000000000	0 0 16																														
		28	1010000000000000	0 0 20																														
		29	1100000000000000	0 0 24																														
		30	1110000000000000	0 0 28																														
		31	1110000000111000	28 0 28																														
	17:13	DataTypeEnum																																
		<p>Lookup one of 32 21-bit values. That value is used (from MSB to LSB) for the Dst.AddrMode, Dst.HorzStride, Src1.SrcType, Src1.RegFile, Src0.SrcType, Src0.RegFile, Dst.DstType, and Dst.RegFile bit fields.</p> <p>Maps to 63:61, 94:89, 46:35</p>																																
		<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>0</td><td>001000000000000000000001</td><td>r:ud a:ud a:ud <1> dir </td></tr> <tr><td>1</td><td>001000000000000010000000</td><td>a:ud r:ud a:ud <1> dir </td></tr> <tr><td>2</td><td>00100000000000001000001</td><td>r:ud r:ud a:ud <1> dir </td></tr> <tr><td>3</td><td>0010000000000011000001</td><td>r:ud i:ud a:ud <1> dir </td></tr> <tr><td>4</td><td>001000000000101011101</td><td>r:f r:d a:ud <1> dir </td></tr> <tr><td>5</td><td>001000000010111011101</td><td>r:f i:vf a:ud <1> dir </td></tr> <tr><td>6</td><td>001000000011101000001</td><td>r:ud r:f a:ud <1> dir </td></tr> <tr><td>7</td><td>001000000011101000101</td><td>r:d r:f a:ud <1> dir </td></tr> <tr><td>8</td><td>001000000011101011101</td><td>r:f r:f a:ud <1> dir </td></tr> </tbody> </table>			Value	Name	Description	0	001000000000000000000001	r:ud a:ud a:ud <1> dir	1	001000000000000010000000	a:ud r:ud a:ud <1> dir	2	00100000000000001000001	r:ud r:ud a:ud <1> dir	3	0010000000000011000001	r:ud i:ud a:ud <1> dir	4	001000000000101011101	r:f r:d a:ud <1> dir	5	001000000010111011101	r:f i:vf a:ud <1> dir	6	001000000011101000001	r:ud r:f a:ud <1> dir	7	001000000011101000101	r:d r:f a:ud <1> dir	8	001000000011101011101	r:f r:f a:ud <1> dir
Value	Name	Description																																
0	001000000000000000000001	r:ud a:ud a:ud <1> dir																																
1	001000000000000010000000	a:ud r:ud a:ud <1> dir																																
2	00100000000000001000001	r:ud r:ud a:ud <1> dir																																
3	0010000000000011000001	r:ud i:ud a:ud <1> dir																																
4	001000000000101011101	r:f r:d a:ud <1> dir																																
5	001000000010111011101	r:f i:vf a:ud <1> dir																																
6	001000000011101000001	r:ud r:f a:ud <1> dir																																
7	001000000011101000101	r:d r:f a:ud <1> dir																																
8	001000000011101011101	r:f r:f a:ud <1> dir																																

EU_INSTRUCTION_COMPACT_TWO_SRC

		9	001000001000001000001	r:ud r:ud r:ud <1> dir
		10	001000011000001000000	a:ud r:ud i:ud <1> dir
		11	001000011000001000001	r:ud r:ud i:ud <1> dir
		12	001000101000101000101	r:d r:d r:d <1> dir
		13	001000111000101000100	a:d r:d i:d <1> dir
		14	001000111000101000101	r:d r:d i:d <1> dir
		15	001011100011101011101	r:f r:f a:f <1> dir
		16	001011101011100011101	r:f a:f r:f <1> dir
		17	001011101011101011100	a:f r:f r:f <1> dir
		18	001011101011101011101	r:f r:f r:f <1> dir
		19	001011111011101011100	a:f r:f i:f <1> dir
		20	000000000010000001100	a:w a:ub a:ud <0> dir
		21	001000000000001011101	r:f r:ud a:ud <1> dir
		22	001000000000101000101	r:d r:d a:ud <1> dir
		23	001000001000001000000	a:ud r:ud r:ud <1> dir
		24	001000101000101000100	a:d r:d r:d <1> dir
		25	001000111000100000100	a:d a:d i:d <1> dir
		26	001001001001000001001	r:uw a:uw r:uw <1> dir
		27	001010111011101011101	r:f r:f i:vf <1> dir
		28	001011111011101011101	r:f r:f i:f <1> dir
		29	001001111001101001100	a:w r:w i:w <1> dir
		30	001001001001001001000	a:uw r:uw r:uw <1> dir
		31	001001011001001001000	a:uw r:uw i:uw <1> dir
12:8	<p>ControlIndex</p> <p>Lookup one of 32 19-bit values. That value is used (from MSB to LSB) for the FlagRegNum, FlagSubRegNum, Saturate, ExecSize, PredInv, PredCtrl, ThreadCtrl, QtrCtrl, DepCtrl, MaskCtrl, and AccessMode bit fields.</p> <p>Maps to 33:32, 31, 23:12, 10:9, 34, 8</p>			
	Value	Name	Description	
	0	0000000000000000010	Align1 We (1) f0.0	
	1	0000100000000000000	Align1 (4) f0.0	
	2	00001000000000000001	Align16 (4) f0.0	
	3	0000100000000000010	Align1 We (4) f0.0	
	4	0000100000000000011	Align16 We (4) f0.0	
	5	0000100000000000100	Align1 NoDDClr (4) f0.0	
	6	0000100000000000101	Align16 NoDDClr (4) f0.0	

EU_INSTRUCTION_COMPACT_TWO_SRC

7	00001000000000000111	Align16 We NoDDClr (4) f0.0
8	00001000000000001000	Align1 NoDDChk (4) f0.0
9	00001000000000001001	Align16 NoDDChk (4) f0.0
10	00001000000000001101	Align16 NoDDClr, NoDDChk (4) f0.0
11	00001100000000000000	Align1 Q1 (8) f0.0
12	00001100000000000001	Align16 Q1 (8) f0.0
13	00001100000000000010	Align1 We Q1 (8) f0.0
14	00001100000000000011	Align16 We Q1 (8) f0.0
15	00001100000000000100	Align1 NoDDClr Q1 (8) f0.0
16	00001100000000000101	Align16 NoDDClr Q1 (8) f0.0
17	00001100000000000111	Align16 We NoDDClr Q1 (8) f0.0
18	00001100000000001001	Align16 NoDDChk Q1 (8) f0.0
19	00001100000000001101	Align16 NoDDClr, NoDDChk Q1 (8) f0.0
20	00001100000000010000	Align1 Q2 (8) f0.0
21	00001100001000000000	Align1 Q1 +f.xyzw (8) f0.0
22	00010000000000000000	Align1 H1 (16) f0.0
23	00010000000000000010	Align1 We H1 (16) f0.0
24	00010000000000000100	Align1 NoDDClr H1 (16) f0.0
25	00010000001000000000	Align1 H1 +f.xyzw (16) f0.0
26	00101100000000000000	Align1 Q1 (8) .sat f0.0
27	00101100000000001000	Align1 Q2 (8) .sat f0.0
28	00110000000000000000	Align1 H1 (16) .sat f0.0
29	00110000001000000000	Align1 H1 +f.xyzw (16) .sat f0.0
30	01010000000000000000	Align1 H1 (16) f0.1
31	01010000001000000000	Align1 H1 +f.xyzw (16) f0.1
7	Reserved	
6:0	Opcode	

EU_INSTRUCTION_CONTROLS_A

EU_INSTRUCTION_CONTROLS_A													
Source:	Eulsa												
Size (in bits):	16												
Default Value:	0x00000000												
DWord	Bit	Description											
0	15:13	<p>ExecSize</p> <table border="1"> <tr> <td>Format:</td> <td>ExecSize</td> </tr> </table> <p>This field determines the number of channels operating in parallel for this instruction. The size cannot exceed the maximum number of channels allowed for the given data type.</p>	Format:	ExecSize									
	Format:	ExecSize											
	12	<p>PredInv</p> <table border="1"> <tr> <td>Exists If:</td> <td>(Property[Predication]== 'true')</td> </tr> </table> <p>This field, together with PredCtrl, enables and controls the generation of the predication mask for the instruction. When it is set, the predication uses the inverse of the predication bits generated according to setting of Predicate Control. In other words, effect of PredInv happens after PredCtrl. This field is ignored by hardware if Predicate Control is set to 0000 - there is no predication. PMask is the final predication mask produced by the effects of both fields.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Positive [Default]</td> <td>Positive polarity of predication. Use the predication mask produced by PredCtrl</td> </tr> <tr> <td>1</td> <td>Negative</td> <td>Negative polarity of predication. If PredCtrl is nonzero, invert the predication mask.</td> </tr> </tbody> </table>	Exists If:	(Property[Predication]== 'true')	Value	Name	Description	0	Positive [Default]	Positive polarity of predication. Use the predication mask produced by PredCtrl	1	Negative	Negative polarity of predication. If PredCtrl is nonzero, invert the predication mask.
	Exists If:	(Property[Predication]== 'true')											
	Value	Name	Description										
0	Positive [Default]	Positive polarity of predication. Use the predication mask produced by PredCtrl											
1	Negative	Negative polarity of predication. If PredCtrl is nonzero, invert the predication mask.											
12	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>(Property[Predication]== 'false')</td> </tr> </table>	Exists If:	(Property[Predication]== 'false')										
Exists If:	(Property[Predication]== 'false')												
11:8	<p>PredCtrl</p> <table border="1"> <tr> <td>Exists If:</td> <td>(Property[Predication]== 'true')</td> </tr> <tr> <td>Format:</td> <td>PredCtrl</td> </tr> </table> <p>This field, together with PredInv, enables and controls the generation of the predication mask for the instruction. It allows per-channel conditional execution of the instruction based on the content of the selected flag register. Encoding depends on the access mode. In Align16 access mode, there are eight encodings (including no predication). All encodings are based on group-of-4 predicate bits, including channel sequential, replication swizzles and horizontal any/all operations. The same configuration is repeated for each group-of-4 execution channels.</p>	Exists If:	(Property[Predication]== 'true')	Format:	PredCtrl								
Exists If:	(Property[Predication]== 'true')												
Format:	PredCtrl												
11:8	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>(Property[Predication]== 'false')</td> </tr> <tr> <td>Format:</td> <td>PredCtrl</td> </tr> </table>	Exists If:	(Property[Predication]== 'false')	Format:	PredCtrl								
Exists If:	(Property[Predication]== 'false')												
Format:	PredCtrl												

EU_INSTRUCTION_CONTROLS_A										
7:6	<p>Thread Control</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">ThreadCtrl</td> </tr> </table> <p>Thread Control. This field provides explicit control for thread switching. If this field is set to 00b, it is up to the GEN execution units to manage thread switching. This is the normal (and unnamed) mode. In this mode, for example, if the current instruction cannot proceed due to operand dependencies, the EU switches to the next available thread to fill the compute pipe. In another example, if the current instruction is ready to go, however, there is another thread with higher priority that also has an instruction ready, the EU switches to that thread. If this field is set to Switch, a forced thread switch occurs after the current instruction is executed and before the next instruction. In addition, a long delay (longer than the execution pipe latency) is introduced for the current thread. Particularly, the instruction queue of the current thread is flushed after the current instruction is dispatched for execution. Switch is designed primarily as a safety feature in case there are race conditions for certain instructions.</p>	Format:	ThreadCtrl							
Format:	ThreadCtrl									
5:4	<p>QtrCtrl</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">QtrCtrl</td> </tr> </table> <p>Quarter Control. This field provides explicit control for ARF selection. This field combined with NibCtrl and ExecSize determines which channels are used for the ARF registers.</p>	Format:	QtrCtrl							
Format:	QtrCtrl									
3	<p>NibCtrl</p> <p>Nibble Control. This field is used in some instructions along with QtrCtrl. See the description of QtrCtrl below. NibCtrl is only used for SIMD4 instructions with a DF (Double Float) source or destination.</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;">0</td> <td style="text-align: center;">Odd</td> <td>Use an odd 1/8th for DMask/VMask and ARF (first, third, fifth, or seventh depending on QtrCtrl).</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">Even</td> <td>Use an even 1/8th for DMask/VMask and ARF (second, fourth, sixth, or eighth depending on QtrCtrl).</td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p style="text-align: center; margin: 0;">Programming Notes</p> <p>Note that if eighths are given zero-based indices from 0 to 7, then NibCtrl = 0 indicates even indices and NibCtrl = 1 indicates odd indices.</p> </div>	Value	Name	Description	0	Odd	Use an odd 1/8th for DMask/VMask and ARF (first, third, fifth, or seventh depending on QtrCtrl).	1	Even	Use an even 1/8th for DMask/VMask and ARF (second, fourth, sixth, or eighth depending on QtrCtrl).
Value	Name	Description								
0	Odd	Use an odd 1/8th for DMask/VMask and ARF (first, third, fifth, or seventh depending on QtrCtrl).								
1	Even	Use an even 1/8th for DMask/VMask and ARF (second, fourth, sixth, or eighth depending on QtrCtrl).								
2:1	<p>DepCtrl</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">DepCtrl</td> </tr> </table> <p>Destination Dependency Control. This field selectively disables destination dependency check and clear for this instruction. When it is set to 00, normal destination dependency control is performed for the instruction - hardware checks for destination hazards to ensure data integrity. Specifically, destination register dependency check is conducted before the instruction is made ready for execution. After the instruction is executed, the destination register scoreboard will be cleared when the destination operands retire. When bit 10 is set (NoDDClr), the destination register scoreboard will NOT be cleared when the destination operands retire. When bit 11 is set (NoDDChk), hardware does not check for destination register dependency before the instruction is made ready for execution. NoDDClr and NoDDChk are not mutual exclusive. When this field is not all-zero, hardware does not protect against destination hazards for the instruction. This is</p>	Format:	DepCtrl							
Format:	DepCtrl									

EU_INSTRUCTION_CONTROLS_A							
	typically used to assemble data in a fine grained fashion (e.g. matrix-vector compute with dot-product instructions), where the data integrity is guaranteed by software based on the intended usage of instruction sequences.						
0	<p>AccessMode Access Mode. This field determines the operand access for the instruction. It applies to all source and destination operands. When it is cleared (Align1), the instruction uses byte-aligned addressing for source and destination operands. Source swizzle control and destination mask control are not supported. When it is set (Align16), the instruction uses 16-byte-aligned addressing for all source and destination operands. Source swizzle control and destination mask control are supported in this 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;">0</td> <td>Align1 [Default]</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Align16</td> </tr> </tbody> </table>	Value	Name	0	Align1 [Default]	1	Align16
Value	Name						
0	Align1 [Default]						
1	Align16						

EU_INSTRUCTION_CONTROLS_B

EU_INSTRUCTION_CONTROLS_B													
Source:	Eulsa												
Size (in bits):	4												
Default Value:	0x00000000												
DWord	Bit	Description											
0	3	Saturate <table border="1"> <tr> <td>Exists If:</td> <td>(Property[Saturation]='true')</td> </tr> </table> <p>Enables or disables destination saturation. When it is set, output values to the destination register are saturated. The saturation operation depends on the destination data type. Saturation is the operation that converts any value outside the saturation target range for the data type to the closest value in the target range. For a floating-point destination type, the saturation target range is [0.0, 1.0]. For a floating-point NaN, there is no <i>closest value</i>; any NaN saturates to 0.0. Note that enabling Saturate overrides all of the NaN propagation behaviors described for various numeric instructions. Any floating-point number greater than 1.0, including +INF, saturates to 1.0. Any negative floating-point number, including -INF, saturates to 0.0. Any floating-point number in the range 0.0 to 1.0 is not changed by saturation. For an integer destination type, the maximum range for that type is the saturation target range. For example, the saturation range for B (Signed Byte Integer) is [-128, 127]. When Saturate is clear, destination values are not saturated. For example, a wrapped result (modulo) is output to the destination for an overflowed integer value. See the Numeric Data Types section for information about data types and their ranges.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No destination modification [Default]</td> <td></td> </tr> <tr> <td>1</td> <td>sat</td> <td>Saturate the output</td> </tr> </tbody> </table>	Exists If:	(Property[Saturation]='true')	Value	Name	Description	0	No destination modification [Default]		1	sat	Saturate the output
		Exists If:	(Property[Saturation]='true')										
		Value	Name	Description									
		0	No destination modification [Default]										
1	sat	Saturate the output											
3	Reserved	<table border="1"> <tr> <td>Exists If:</td> <td>(Property[Saturation]='false')</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	(Property[Saturation]='false')	Format:	MBZ							
		Exists If:	(Property[Saturation]='false')										
Format:	MBZ												
2	Reserved												
1	CmptCtrl	<p>Compaction Control Indicates whether the instruction is compacted to the 64-bit compact instruction format. When this bit is set, the 64-bit compact instruction format is used. The EU decodes the compact format using lookup tables internal to the hardware, but documented for use by software tools. Only some instruction variations can be compacted, chv the variations supported by those lookup tables and the compact format. See EU Compact Instruction Format for more information.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>NoCompaction</td> <td>No compaction. 128-bit native instruction supporting all instruction options.</td> </tr> <tr> <td>1</td> <td>Compacted</td> <td>Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.</td> </tr> </tbody> </table>	Value	Name	Description	0	NoCompaction	No compaction. 128-bit native instruction supporting all instruction options.	1	Compacted	Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.		
		Value	Name	Description									
		0	NoCompaction	No compaction. 128-bit native instruction supporting all instruction options.									
1	Compacted	Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.											

EU_INSTRUCTION_CONTROLS_B		
0	AccWrCtrl AccWrCtrl. This field allows per instruction accumulator write control.	
Value	Name	Description
0	Don't write to ACC [Default]	
1	Update ACC	Write result to the ACC, and destination

EU_INSTRUCTION_CONTROLS

EU_INSTRUCTION_CONTROLS		
Source:	Eulsa	
Size (in bits):	24	
Default Value:	0x00000000	
DWord	Bit	Description
0	23:20	Controls B
		Format: EU_INSTRUCTION_CONTROLS_B
	19:16	Reserved
		Exists If: (Property[Conditional Modifier]='false')
		Format: MBZ
	19:16	CondModifier
		Exists If: (Property[Conditional Modifier]='true')
		Format: CondModifier
		Does not exist for send/sendc/math/branch/break-continue opcodes
	15:0	Controls A
Format: EU_INSTRUCTION_CONTROLS_A		

EU_INSTRUCTION_HEADER

EU_INSTRUCTION_HEADER		
Source:	Eulsa	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:8	Control Format: EU_INSTRUCTION_CONTROLS
	7	Reserved Format: MBZ
	6:0	Opcode Format: EU_OPCODE



EU_INSTRUCTION_ILLEGAL

EU_INSTRUCTION_ILLEGAL		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..3	127:7	Reserved Format: MBZ
	6:0	Opcode Format: EU_OPCODE

EU_INSTRUCTION_MATH

EU_INSTRUCTION_MATH		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
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
	6:0	Opcode Format: EU_OPCODE

EU_INSTRUCTION_NOP

EU_INSTRUCTION_NOP		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..3	127:31	Reserved Format: MBZ
	30	Reserved
	29:7	Reserved Format: MBZ
	6:0	Opcode Format: EU_OPCODE

EU_INSTRUCTION_OPERAND_CONTROLS

EU_INSTRUCTION_OPERAND_CONTROLS		
Source:	Eulsa	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:16	Destination Register Region
		Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]== 'Align16')
		Format: EU_INSTRUCTION_OPERAND_DST_ALIGN16
	31:16	Destination Register Region
		Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]== 'Align1')
		Format: EU_INSTRUCTION_OPERAND_DST_ALIGN1
	15	Reserved
		Exists If: ([Destination Register Region][Destination Addressing Mode]== 'Direct')
		Format: MBZ
	15	Destination Address Immediate[9:9]
Exists If: ([Destination Register Region][Destination Addressing Mode]== 'Indirect')		
	Format: U1	
14:11	Src0.SrcType	
	Exists If: ([Src0.RegFile]!= 'IMM')	
	Format: SrcType	
14:11	Src0.SrcType	
	Exists If: ([Src0.RegFile]= 'IMM')	
	Format: SrcImmType	
10:9	Src0.RegFile	
	Format: RegFile	
8:5	Destination Data Type	
	Format: DstType	
This field specifies the numeric data type of the destination operand dst. The bits of the destination operand are interpreted as the identified numeric data type, rather than coerced into a type implied by the operator. For a send instruction, this field applies to the CurrDst - the current destination operand.		
4:3	Destination Register File	
	Format: RegFile	

EU_INSTRUCTION_OPERAND_CONTROLS											
2	<p>MaskCtrl Mask Control (formerly Write Enable Control). This field determines if the the per channel write enables are used to generate the final write enable. This field should be normally "0".</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>Normal [Default]</td> <td></td> </tr> <tr> <td style="text-align: center;">1</td> <td>Write all channels</td> <td>Except channels killed with predication control</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>MaskCtrl = NoMask skips the check for PciP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.</p>		Value	Name	Description	0	Normal [Default]		1	Write all channels	Except channels killed with predication control
Value	Name	Description									
0	Normal [Default]										
1	Write all channels	Except channels killed with predication control									
1:0	Flag Register Number/Subregister Number										

EU_INSTRUCTION_OPERAND_DST_ALIGN1

EU_INSTRUCTION_OPERAND_DST_ALIGN1						
Source:	Eulsa					
Size (in bits):	16					
Default Value:	0x00000000					
DWord	Bit	Description				
0	15	<p>Destination Addressing Mode</p> <table border="1"> <tr> <td>Format:</td> <td>AddrMode</td> </tr> </table> <p>For a send instruction, this field applies to PostDst - the post destination operand. Addressing mode for CurrDst (current destination operand) is fixed as Direct. (See Instruction Reference chapter for CurrDst and PostDst.)</p>	Format:	AddrMode		
	Format:	AddrMode				
	14:13	<p>Destination Horizontal Stride</p> <table border="1"> <tr> <td>Format:</td> <td>HorzStride</td> </tr> </table> <p>For a send instruction, this field applies to CurrDst. PostDst only uses the register number.</p>	Format:	HorzStride		
	Format:	HorzStride				
	12:9	<p>Destination Address Subregister Number</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Destination Addressing Mode])=='Indirect')</td> </tr> <tr> <td>Format:</td> <td>AddrSubRegNum</td> </tr> </table> <p>For a send instruction, this field applies to PostDst</p>	Exists If:	(([Destination Addressing Mode])=='Indirect')	Format:	AddrSubRegNum
	Exists If:	(([Destination Addressing Mode])=='Indirect')				
Format:	AddrSubRegNum					
12:5	<p>Destination Register Number</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Destination Addressing Mode])=='Direct')</td> </tr> <tr> <td>Format:</td> <td>DstRegNum</td> </tr> </table> <p>For a send instruction, this field applies to PostDst.</p>	Exists If:	(([Destination Addressing Mode])=='Direct')	Format:	DstRegNum	
Exists If:	(([Destination Addressing Mode])=='Direct')					
Format:	DstRegNum					
8:0	<p>Destination Address Immediate</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Destination Addressing Mode])=='Indirect')</td> </tr> <tr> <td>Format:</td> <td>S8</td> </tr> </table> <p>For a send instruction, this field applies to PostDst.</p>	Exists If:	(([Destination Addressing Mode])=='Indirect')	Format:	S8	
Exists If:	(([Destination Addressing Mode])=='Indirect')					
Format:	S8					
4:0	<p>Destination Subregister Number</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Destination Addressing Mode])=='Direct')</td> </tr> <tr> <td>Format:</td> <td>DstSubRegNum</td> </tr> </table> <p>For a send instruction, this field applies to CurrDst.</p>	Exists If:	(([Destination Addressing Mode])=='Direct')	Format:	DstSubRegNum	
Exists If:	(([Destination Addressing Mode])=='Direct')					
Format:	DstSubRegNum					

EU_INSTRUCTION_OPERAND_DST_ALIGN16

EU_INSTRUCTION_OPERAND_DST_ALIGN16						
Source:	Eulsa					
Size (in bits):	16					
Default Value:	0x00000000					
DWord	Bit	Description				
0	15	<p>Destination Addressing Mode</p> <table border="1"> <tr> <td>Format:</td> <td>AddrMode</td> </tr> </table> <p>For a send instruction, this field applies to PostDst - the post destination operand. Addressing mode for CurrDst (current destination operand) is fixed as Direct. (See Instruction Reference chapter for CurrDst and PostDst.)</p>	Format:	AddrMode		
	Format:	AddrMode				
	14:13	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	12:9	<p>Destination Address Subregister Number</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Destination Addressing Mode]='Indirect')</td> </tr> <tr> <td>Format:</td> <td>AddrSubRegNum</td> </tr> </table> <p>For a send instruction, this field applies to PostDst</p>	Exists If:	(([Destination Addressing Mode]='Indirect')	Format:	AddrSubRegNum
	Exists If:	(([Destination Addressing Mode]='Indirect')				
	Format:	AddrSubRegNum				
12:5	<p>Destination Register Number</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Destination Addressing Mode]='Direct')</td> </tr> <tr> <td>Format:</td> <td>DstRegNum</td> </tr> </table> <p>For a send instruction, this field applies to PostDst.</p>	Exists If:	(([Destination Addressing Mode]='Direct')	Format:	DstRegNum	
Exists If:	(([Destination Addressing Mode]='Direct')					
Format:	DstRegNum					
8:4	<p>Destination Address Immediate[8:4]</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Destination Addressing Mode]='Indirect')</td> </tr> <tr> <td>Format:</td> <td>S8[8:4]</td> </tr> </table> <p>For a send instruction, this field applies to PostDst</p>	Exists If:	(([Destination Addressing Mode]='Indirect')	Format:	S8[8:4]	
Exists If:	(([Destination Addressing Mode]='Indirect')					
Format:	S8[8:4]					
4	<p>Destination Subregister Number</p> <table border="1"> <tr> <td>Exists If:</td> <td>(([Destination Addressing Mode]='Direct')</td> </tr> <tr> <td>Format:</td> <td>DstSubRegNum[4:4]</td> </tr> </table> <p>For a send instruction, this field applies to CurrDst.</p>	Exists If:	(([Destination Addressing Mode]='Direct')	Format:	DstSubRegNum[4:4]	
Exists If:	(([Destination Addressing Mode]='Direct')					
Format:	DstSubRegNum[4:4]					
3:0	<p>Destination Channel Enable</p> <table border="1"> <tr> <td>Format:</td> <td>ChanEn[4]</td> </tr> </table> <p>For a send instruction, this field applies to the CurrDst</p>	Format:	ChanEn[4]			
Format:	ChanEn[4]					

EU_INSTRUCTION_OPERAND_SEND_MSG

EU_INSTRUCTION_OPERAND_SEND_MSG								
Source:	Eulsa							
Size (in bits):	32							
Default Value:	0x00000000							
DWord	Bit	Description						
0	31	EOT						
		Description						
		This field controls the termination of the thread. For a send instruction, if this field is set, EU will terminate the thread and also set the EOT bit in the message sideband. This field only applies to the send instruction. It is not present for other instructions.						
		<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>0</td> <td>Thread is not terminated</td> </tr> <tr> <td>1</td> <td>EOT</td> </tr> </tbody> </table>		Value	Name	0	Thread is not terminated	1
	Value	Name						
	0	Thread is not terminated						
	1	EOT						
	30:29	Reserved						
		Format:	MBZ					
	30:0	Message Descriptor						
Exists If:		[SelReg32Desc]='IMM'						
30:0	Format:	MsgDescpt31						
	Reg32							
30:0	Exists If:	[SelReg32Desc]!='IMM'						
	In a send or sendc instruction refers to the option of providing the message descriptor field DWord, of which bits 30:0 are used, in the first two words of the Address Register rather than as an immediate operand.							
28:0	Message Descriptor							
	Exists If:	(Structure[EU_INSTRUCTION_SEND][Src1.RegFile]='IMM')						
28:0	Format:	MsgDescpt31						
	Reg32							
28:0	Exists If:	(Structure[EU_INSTRUCTION_SEND][Src1.RegFile]!='IMM')						
	In a send or sendc instruction refers to the option of providing the message descriptor field DWord, of which bits 28:0 are used, in the first two words of the Address Register rather than as an immediate operand.							

EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1

EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1		
Source:	Eulsa	
Size (in bits):	25	
Default Value:	0x00000000	
DWord	Bit	Description
0	24:21	Source Vertical Stride
		Format: VertStride
	20:18	Source Width
		Format: Width
	17:16	Source Horizontal Stride
		Format: HorzStride
	15	Source Addressing Mode
		Format: AddrMode
	14:13	Source Modifier
		Exists If: (Property[Source Modifier] == 'true')
	14:13	Reserved
		Exists If: (Property[Source Modifier] == 'false')
12:9	Source Address Subregister Number	
	Exists If: (([Source Addressing Mode] == 'Indirect')	
12:5	Source Register Number	
	Exists If: (([Source Addressing Mode] == 'Direct')	
8:0	Source Address Immediate [8:0]	
	Exists If: (([Source Addressing Mode] == 'Indirect')	
4:0	Source Subregister Number	
	Exists If: (([Source Addressing Mode] == 'Direct')	
		Format: SrcSubRegNum

EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16

EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16		
Source:	Eulsa	
Size (in bits):	25	
Default Value:	0x00000000	
DWord	Bit	Description
0	24:21	Source Vertical Stride Format: VertStride
	20	Reserved Format: MBZ
	19:16	Source Channel Select[7:4] Format: ChanSel[4][7:4]
	15	Source Addressing Mode Format: AddrMode
	14:13	Source Modifier Exists If: (Property[Source Modifier]== 'true') Format: SrcMod
	14:13	Reserved Exists If: (Property[Source Modifier]== 'false') Format: MBZ
	12:9	Source Address Subregister Number Exists If: ([Source Addressing Mode]== 'Indirect') Format: AddrSubRegNum
	12:5	Source Register Number Exists If: ([Source Addressing Mode]== 'Direct') Format: SrcRegNum
	8:4	Source Address Immediate[8:4] Exists If: ([Source Addressing Mode]== 'Indirect') Format: S9[8:4]
	4	Source Subregister Number[4:4] Exists If: ([Source Addressing Mode]== 'Direct') Format: SrcSubRegNum[4:4]
	3:0	Source Channel Select[3:0] Format: ChanSel[4][3:0]

EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC

EU_INSTRUCTION_OPERAND_SRC_REG_THREE_SRC		
Source:	Eulsa	
Size (in bits):	21	
Default Value:	0x00000000	
DWord	Bit	Description
0	20	Source Subregister Number [1] Format: SrcSubRegNum[1]
	19:12	Source Register Number Format: SrcRegNum
	11:9	Source Subregister Number [4:2] Format: SrcSubRegNum[4:2]
	8:1	Source Swizzle Format: ChanSel[4]
	0	Source Replicate Control Format: RepCtrl

EU_INSTRUCTION_SEND

EU_INSTRUCTION_SEND		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..3	127:96	Message Format: EU_INSTRUCTION_OPERAND_SEND_MSG
	95	Reserved Format: MBZ
	94:91	ExDesc[31:28] Format: ExtMsgDescpt[31:28]
	90:89	Reserved Format: MBZ
	88:85	ExDesc[27:24] Format: ExtMsgDescpt[27:24]
	84	Reserved Format: MBZ
	83:80	ExDesc[23:20] Format: ExtMsgDescpt[23:20]
	79:68	Reserved Format: MBZ
	67:64	ExDesc[19:16] Format: ExtMsgDescpt[19:16]
	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

EU_INSTRUCTION_SENDS

EU_INSTRUCTION_SENDS		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
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
	62	Destination Address Immediate Sign [9] Exists If: ([Destination Addressing Mode]='Indirect') Format: S9[9]

EU_INSTRUCTION_SENDS		
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
6:0	Opcode	
	Format:	EU_OPCODE

EU_INSTRUCTION_SOURCES_IMM32

EU_INSTRUCTION_SOURCES_IMM32			
Source:	Eulsa		
Size (in bits):	64		
Default Value:	0x00000000, 0x00000000		
Single source, immediate			
DWord	Bit	Description	
0..1	63:32	Source 0 Immediate	
	31:25	Reserved	
		Format:	MBZ
	24:0	Source 0	
		Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16') AND (Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')
		Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
	24:0	Source 0	
		Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1') AND (Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')
	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1	

EU_INSTRUCTION_SOURCES_REG

EU_INSTRUCTION_SOURCES_REG		
Source:	Eulsa	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
Single source, register		
DWord	Bit	Description
0..1	63:25	Reserved
		Format: MBZ
	24:0	Source 0
		Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16') AND (Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')
		Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
	24:0	Source 0
Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1') AND (Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')		
Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1		

EU_INSTRUCTION_SOURCES_REG_IMM

EU_INSTRUCTION_SOURCES_REG_IMM			
Source:	Eulsa		
Size (in bits):	64		
Default Value:	0x00000000, 0x00000000		
Dual source, register and immediate			
DWord	Bit	Description	
0..1	63:32	Source 1 Immediate	
	31	Reserved	
		Exists If:	(([Source 0][Source Addressing Mode])=='Direct')
		Format:	MBZ
	31	Source 0 Address Immediate [9] (Sign Bit)	
		Exists If:	(([Source 0][Source Addressing Mode])=='Indirect')
		Format:	S9[9]
	30:27	Src1.SrcType	
		Format:	SrcImmType
	26:25	Src1.RegFile	
Format:		RegFile	
24:0	Source 0		
	Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]!='Align16') AND (Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')	
	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16	
24:0	Source 0		
	Exists If:	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]!='Align1') AND (Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')	
	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1	

EU_INSTRUCTION_SOURCES_REG_REG

EU_INSTRUCTION_SOURCES_REG_REG		
Source:	Eulsa	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
Dual source, both registers		
DWord	Bit	Description
0..1	63:58	Reserved Format: MBZ
	57	Reserved Exists If: ([Source 1][Source Addressing Mode]== 'Direct') Format: MBZ
	57	Source 1 Address Immediate [9] (Sign Bit) Exists If: ([Source 1][Source Addressing Mode]== 'Indirect') Format: S9[9]
	56:32	Source 1 Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]== 'Align16') Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
	56:32	Source 1 Exists If: (Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]== 'Align1') Format: EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1
	31	Reserved Exists If: ([Source 0][Source Addressing Mode]== 'Direct') Format: MBZ
	31	Source 0 Address Immediate [9] (Sign Bit) Exists If: ([Source 0][Source Addressing Mode]== 'Indirect') Format: S9[9]

EU_INSTRUCTION_SOURCES_REG_REG							
30:27	<p>Src1.SrcType</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">SrcType</td> </tr> </table> <p>This field specifies the numeric data type of the source operand src1. The bits of a source operand are interpreted as the identified numeric data type, rather than coerced into a type implied by the operator. Depending on RegFile field of the source operand, there are two different encoding for this field. If a source is a register operand, this field follows the Source Register Type Encoding. If a source is an immediate operand, this field follows the Source Immediate Type Encoding.</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>11b</td> <td>Reserved</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Both source operands, src0 and src1, support immediate types, but only one immediate is allowed for a given instruction and it must be the last operand.</p> <p>Halfbyte integer vector (v) type can only be used in instructions in packed-word execution mode. Therefore, in a two-source instruction where src1 is of type :v, src0 must be of type :b, :ub, :w, or :uw.</p>	Format:	SrcType	Value	Name	11b	Reserved
Format:	SrcType						
Value	Name						
11b	Reserved						
26:25	<p>Src1.RegFile</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">RegFile</td> </tr> </table>	Format:	RegFile				
Format:	RegFile						
24:0	<p>Source 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists</td> <td>(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16') AND</td> </tr> <tr> <td>If:</td> <td>(Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16</td> </tr> </table>	Exists	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16') AND	If:	(Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16
Exists	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align16') AND						
If:	(Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')						
Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN16						
24:0	<p>Source 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists</td> <td>(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1') AND</td> </tr> <tr> <td>If:</td> <td>(Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')</td> </tr> <tr> <td>Format:</td> <td>EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1</td> </tr> </table>	Exists	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1') AND	If:	(Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')	Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1
Exists	(Structure[EU_INSTRUCTION_CONTROLS_A][AccessMode]='Align1') AND						
If:	(Structure[EU_INSTRUCTION_OPERAND_CONTROLS][Src0.RegFile]!='IMM')						
Format:	EU_INSTRUCTION_OPERAND_SRC_REG_ALIGN1						

Execution_Unit_Extended_Message_Descriptor

Execution_Unit_Extended_Message_Descriptor							
Source:	BSpec						
Size (in bits):	11						
Default Value:	0x00000000						
DWord	Bit	Description					
0	10	Reserved Format: MBZ					
	9:6	Extended Message Length Exists If: <code>Structure[EU_INSTRUCTION_HEADER][Opcode]== 'Sends' OR Structure[EU_INSTRUCTION_HEADER][Opcode]== 'Sendsc'</code> Format: U4 This field specifies the number of 256-bit GRF registers starting from <src1> to be sent out on the request message payload. Valid value ranges from 0 to 15. Must be 0 when <src1> is null register. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,15]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,15]		
	Value	Name					
	[0,15]						
	9:6	Reserved Exists If: <code>Structure[EU_INSTRUCTION_HEADER][Opcode]== 'Send' OR Structure[EU_INSTRUCTION_HEADER][Opcode]== 'Sendc'</code> Format: MBZ					
	5	End Of Thread This field, if set, indicates that this is the final message of the thread and the thread's resources can be reclaimed. This bit maps to bit 127 of the send/sends instruction. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No Termination</td> </tr> <tr> <td>1</td> <td>EOT</td> </tr> </tbody> </table>	Value	Name	0	No Termination	1
Value	Name						
0	No Termination						
1	EOT						
4	Reserved Format: MBZ						
3:0	Target Function ID This field indicates the function unit for which the message is intended. Refer to "GPU Overview" document for the mapping of Shared Function IDs						

Extended Message Descriptor - Execution Unit

Extended Message Descriptor - Execution Unit								
Source:	BSpec							
Size (in bits):	32							
Default Value:	0x00000000							
DWord	Bit	Description						
	31:16	<p>Extended Function Control</p> <table border="1"> <tr> <td>Exists</td> <td>//(Structure[EU_INSTRUCTION_HEADER][Opcode]!='Sel') OR</td> </tr> <tr> <td>If:</td> <td>([SelReg32ExDesc]='IMM')</td> </tr> <tr> <td>Format:</td> <td>U16</td> </tr> </table>	Exists	//(Structure[EU_INSTRUCTION_HEADER][Opcode]!='Sel') OR	If:	([SelReg32ExDesc]='IMM')	Format:	U16
Exists	//(Structure[EU_INSTRUCTION_HEADER][Opcode]!='Sel') OR							
If:	([SelReg32ExDesc]='IMM')							
Format:	U16							
	31:12	<p>Extended Function Control</p> <table border="1"> <tr> <td>Exists</td> <td>//(Structure[EU_INSTRUCTION_HEADER][Opcode]='Sel') AND</td> </tr> <tr> <td>If:</td> <td>([SelReg32ExDesc]!='IMM')</td> </tr> <tr> <td>Format:</td> <td>U20</td> </tr> </table> <p>This field is intended to control the target function unit. Refer to the section on the specific target function unit for details on the contents of this field.</p>	Exists	//(Structure[EU_INSTRUCTION_HEADER][Opcode]='Sel') AND	If:	([SelReg32ExDesc]!='IMM')	Format:	U20
Exists	//(Structure[EU_INSTRUCTION_HEADER][Opcode]='Sel') AND							
If:	([SelReg32ExDesc]!='IMM')							
Format:	U20							
	15:12	<p>Reserved</p> <table border="1"> <tr> <td>Exists</td> <td>//(Structure[EU_INSTRUCTION_HEADER][Opcode]!='Sel') OR</td> </tr> <tr> <td>If:</td> <td>([SelReg32ExDesc]='IMM')</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists	//(Structure[EU_INSTRUCTION_HEADER][Opcode]!='Sel') OR	If:	([SelReg32ExDesc]='IMM')	Format:	MBZ
Exists	//(Structure[EU_INSTRUCTION_HEADER][Opcode]!='Sel') OR							
If:	([SelReg32ExDesc]='IMM')							
Format:	MBZ							
	11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ							
	10:0	<p>Execution Unit Extended Message Descriptor Definition</p> <table border="1"> <tr> <td>Format:</td> <td>Execution_Unit_Extended_Message_Descriptor</td> </tr> </table>	Format:	Execution_Unit_Extended_Message_Descriptor				
Format:	Execution_Unit_Extended_Message_Descriptor							

Extended Message Descriptor Render Target

Extended Message Descriptor Render Target		
Source:	BSpec	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:20	Reserved Format: MBZ
	15:12	Reserved Format: MBZ
	9:6	Extended Message Length Format: U4 This field specifies the number of 256-bit GRF registers starting from <src1> to be sent out on the request message payload. Valid value ranges from 0 to 15. Must be 0 when <src1> is null register.
	5	End of Thread This field, if set, indicates that this is the final message of the thread and the thread's resources can be reclaimed.
	4	Reserved Format: MBZ
	3:0	Target Function ID This field indicates the function unit for which the message is intended. <i>Refer to "GPU Overview" document for the mapping of Shared Function IDs</i>

Extended Message Descriptor - Sampling Engine

Extended Message Descriptor - Sampling Engine				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:12	<p>Bindless Surface Offset</p> <table border="1"> <tr> <td>Format:</td> <td>BindlessSurfaceOffset[25:6]</td> </tr> </table> <p>Specifies the bindless surface offset if the Binding Table Index is set to 252. Ignored otherwise. The bindless surface offset is added to the Bindless Surface Base Address as bits 25:6 of the byte-based address. The resulting address is the location of SURFACE_STATE for this message.</p>	Format:	BindlessSurfaceOffset[25:6]
	Format:	BindlessSurfaceOffset[25:6]		
	11	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
10:0	<p>Execution Unit Extended Message Descriptor Definition</p> <table border="1"> <tr> <td>Format:</td> <td>Execution_Unit_Extended_Message_Descriptor</td> </tr> </table>	Format:	Execution_Unit_Extended_Message_Descriptor	
Format:	Execution_Unit_Extended_Message_Descriptor			

ExtMsgDescpt

ExtMsgDescpt																	
Source:	Eulsa																
Size (in bits):	32																
Default Value:	0x00000000																
DWord	Bit	Description															
0 Extended Message Descriptor Definition for SendS (Immediate)	31:16	Extended Function Control Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">U16</td></tr></table> This field is intended to control the target function unit. Refer to the section on the specific target function unit for details on the contents of this field.		U16													
		U16															
	15:12	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	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															
	10:6	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															
5	EOT Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">U1</td></tr></table> This field, if set, indicates that this is the final message of the thread and the thread's resources can be reclaimed. <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>No Termination</td> </tr> <tr> <td style="text-align: center;">1</td> <td>EOT</td> </tr> </tbody> </table>		U1	Value	Name	0	No Termination	1	EOT								
	U1																
Value	Name																
0	No Termination																
1	EOT																
4	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																
3:0	Target Function ID Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"></td><td style="text-align: center;">U4</td></tr></table> If set, indicates that the message includes a header. Depending on the target shared function, this field may be restricted to either enabled or disabled. Refer to the specific shared function section for details. <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;">0000b</td> <td>Null</td> </tr> <tr> <td style="text-align: center;">0001b</td> <td>Reserved</td> </tr> <tr> <td style="text-align: center;">0010b</td> <td>SamplingEngine</td> </tr> <tr> <td style="text-align: center;">0011b</td> <td>MessageGateway</td> </tr> <tr> <td style="text-align: center;">0100b</td> <td>DataPortSamplerCache</td> </tr> <tr> <td style="text-align: center;">0101b</td> <td>DataPortRenderCache</td> </tr> </tbody> </table>		U4	Value	Name	0000b	Null	0001b	Reserved	0010b	SamplingEngine	0011b	MessageGateway	0100b	DataPortSamplerCache	0101b	DataPortRenderCache
	U4																
Value	Name																
0000b	Null																
0001b	Reserved																
0010b	SamplingEngine																
0011b	MessageGateway																
0100b	DataPortSamplerCache																
0101b	DataPortRenderCache																

ExtMsgDescpt	
	0110b URB
	0111b ThreadSpawner
	1000b VideoMotionEstimation
	1001b ConstantCache
	1010b-1111b Reserved

ExtMsgDescptImmediate

ExtMsgDescptImmediate													
Source:	Eulsa												
Size (in bits):	32												
Default Value:	0x00000000												
DWord	Bit	Description											
0 Extended Message Descriptor Definition for SendS (Immediate)	31:16	Extended Function Control Format: U16 This field is intended to control the target function unit. Refer to the section on the specific target function unit for details on the contents of this field.											
	15:12	Reserved Format: MBZ											
	11	Reserved Format: MBZ											
	10	Reserved Format: MBZ											
	9:6	Reserved Format: MBZ											
	5	EOT Format: U1 This field, if set, indicates that this is the final message of the thread and the thread's resources can be reclaimed. <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>No Termination</td> </tr> <tr> <td style="text-align: center;">1</td> <td>EOT</td> </tr> </tbody> </table>	Value	Name	0	No Termination	1	EOT					
	Value	Name											
	0	No Termination											
1	EOT												
4	Reserved Format: MBZ												
3:0	Target Function ID Format: U4 If set, indicates that the message includes a header. Depending on the target shared function, this field may be restricted to either enabled or disabled. Refer to the specific shared function section for details. <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;">0000b</td> <td>Null</td> </tr> <tr> <td style="text-align: center;">0001b</td> <td>Reserved</td> </tr> <tr> <td style="text-align: center;">0010b</td> <td>SamplingEngine</td> </tr> <tr> <td style="text-align: center;">0011b</td> <td>MessageGateway</td> </tr> <tr> <td style="text-align: center;">0100b</td> <td>DataPortSamplerCache</td> </tr> </tbody> </table>	Value	Name	0000b	Null	0001b	Reserved	0010b	SamplingEngine	0011b	MessageGateway	0100b	DataPortSamplerCache
Value	Name												
0000b	Null												
0001b	Reserved												
0010b	SamplingEngine												
0011b	MessageGateway												
0100b	DataPortSamplerCache												

ExtMsgDescptImmediate	
	0101b DataPortRenderCache
	0110b URB
	0111b ThreadSpawner
	1000b VideoMotionEstimation
	1001b ConstantCache
	1010b-1111b Reserved

FFTID Message Header Control

MHC_FFTID - FFTID Message Header Control				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:8	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
	Format:	Ignore		
7:0	<p>FFTID</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>U8</td> </tr> </table> <p>Fixed function thread ID, used to free up resources by the thread on thread completion.</p>	Format:	U8	
Format:	U8			



Filter_Coefficient

Filter_Coefficient						
Source:	BSpec					
Size (in bits):	8					
Default Value:	0x00000000					
DWord	Bit	Description				
0	7:0	Filter Coefficient <table border="1"> <tr> <td>Format:</td> <td>S1.6 2's Complement</td> </tr> <tr> <td colspan="2">Range : [-1 63/64, +1 63/64]</td> </tr> </table>	Format:	S1.6 2's Complement	Range : [-1 63/64, +1 63/64]	
Format:	S1.6 2's Complement					
Range : [-1 63/64, +1 63/64]						

Filter_Coefficients

Filter_Coefficients		
Source:	BSpec	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0	63:56	Filter Coefficient Offset 7 Format: <input type="text"/> Filter_Coefficient
	55:48	Filter Coefficient Offset 6 Format: <input type="text"/> Filter_Coefficient
	47:40	Filter Coefficient Offset 5 Format: <input type="text"/> Filter_Coefficient
	39:32	Filter Coefficient Offset 4 Format: <input type="text"/> Filter_Coefficient
	31:24	Filter Coefficient Offset 3 Format: <input type="text"/> Filter_Coefficient
	23:16	Filter Coefficient Offset 2 Format: <input type="text"/> Filter_Coefficient
	15:8	Filter Coefficient Offset 1 Format: <input type="text"/> Filter_Coefficient
	7:0	Filter Coefficient Offset 0 Format: <input type="text"/> Filter_Coefficient

FrameDeltaQp

FrameDeltaQp				
Source:	BSpec			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0..1	63:56	FrameDeltaQp[7] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px; text-align: center;">S7</td></tr></table>		S7
		S7		
	55:48	FrameDeltaQp[6] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px; text-align: center;">S7</td></tr></table>		S7
		S7		
	47:40	FrameDeltaQp[5] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px; text-align: center;">S7</td></tr></table>		S7
		S7		
	39:32	FrameDeltaQp[4] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px; text-align: center;">S7</td></tr></table>		S7
		S7		
31:24	FrameDeltaQp[3] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px; text-align: center;">S7</td></tr></table>		S7	
	S7			
23:16	FrameDeltaQp[2] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px; text-align: center;">S7</td></tr></table>		S7	
	S7			
15:8	FrameDeltaQp[1] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px; text-align: center;">S7</td></tr></table>		S7	
	S7			
7:0	FrameDeltaQp[0] Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px; text-align: center;">S7</td></tr></table>		S7	
	S7			

FrameDeltaQpRange

FrameDeltaQpRange				
Source:	BSpec			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0..1	63:56	FrameDeltaQpRange[7] Format: <table border="1" style="display: inline-table; width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U8</td></tr></table>		U8
		U8		
	55:48	FrameDeltaQpRange[6] Format: <table border="1" style="display: inline-table; width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U8</td></tr></table>		U8
		U8		
	47:40	FrameDeltaQpRange[5] Format: <table border="1" style="display: inline-table; width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U8</td></tr></table>		U8
		U8		
	39:32	FrameDeltaQpRange[4] Format: <table border="1" style="display: inline-table; width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U8</td></tr></table>		U8
		U8		
31:24	FrameDeltaQpRange[3] Format: <table border="1" style="display: inline-table; width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U8</td></tr></table>		U8	
	U8			
23:16	FrameDeltaQpRange[2] Format: <table border="1" style="display: inline-table; width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U8</td></tr></table>		U8	
	U8			
15:8	FrameDeltaQpRange[1] Format: <table border="1" style="display: inline-table; width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U8</td></tr></table>		U8	
	U8			
7:0	FrameDeltaQpRange[0] Format: <table border="1" style="display: inline-table; width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U8</td></tr></table>		U8	
	U8			

FunctionControl

FunctionControl			
Source:	Eulsa		
Size (in bits):	6		
Default Value:	0x00000000		
DWord	Bit	Description	
0	5:4	Reserved	
	3:0	Target Function ID	
		Value	Name
		0000b	Reserved
		0001b	INV (Reciprocal)
		0010b	LOG
		0011b	EXP
		0100b	SQRT
		0101b	RSQ
		0110b	SIN
		0111b	COS
		1000b	Reserved
		1001b	FDIV
		1010b	POW
		1011b	INT DIV Quotient and remainder
		1100b	INT DIV Quotient only
		1101b	INT DIV Remainder only
	1110b	INVM	
	1111b	RSQRTM	

GATHER_CONSTANT_ENTRY

GATHER_CONSTANT_ENTRY					
Source:	RenderCS				
Size (in bits):	16				
Default Value:	0x00000000				
DWord	Bit	Description			
0	15:8	<p>Constant Buffer Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>Offset[7:0]ConstantBuffer</td> </tr> </table> <p>This field specifies the Offset in 128-bit units of the 128b entry fetched from the constant buffer for this entry (including when On-Die Table Read Enable is set).</p>	Format:	Offset[7:0]ConstantBuffer	
	Format:	Offset[7:0]ConstantBuffer			
	7:4	<p>Channel Mask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Mask:</td> <td>Mask[3:0]</td> </tr> <tr> <td>Format:</td> <td>ConstantBuffer</td> </tr> </table> <p>Each bit of this field correspond to the 4 channels of each entry fetched from memory. When the bit is a 1, the corresponding 32-bit value is loaded in FF's push constant buffer. When the bit is a 0, the corresponding 32-bit value is not loaded. If this field is zero it means the entry is not used.</p>	Mask:	Mask[3:0]	Format:
Mask:	Mask[3:0]				
Format:	ConstantBuffer				
3:0	<p>Binding Table Index Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>Constant Buffer Index offset [3:0]Surface State for ConstantBuffer</td> </tr> </table> <p>This field specifies the Binding Table index offset from the Constant Buffer Binding Table Block starting point in the Binding Table. This value is added to the Constant Buffer Binding Table Block will result in the Binding Table Index pointing to the surface state containing the constant buffer to be referenced. If VS Constant Buffer Dx9 Enable is set then a value of '1' specifies that the fetch to the constant buffer should be offset by 4KB in order to address the upper 4K of the constant buffer. Any value greater than '1' is invalid when VS Constant Buffer Dx9 Enable is set.</p>	Format:	Constant Buffer Index offset [3:0]Surface State for ConstantBuffer		
Format:	Constant Buffer Index offset [3:0]Surface State for ConstantBuffer				

GTC Interrupt Bit Definition

GTC Interrupt Bit Definition				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
The GTC Interrupt Registers all share the same bit definitions from this table.				
DWord	Bit	Description		
0	31:26	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
	25	GTC Lock Loss GTC has lost lock with a remote GTC sink. The difference between the local and remote GTC has exceeded programmed threshold.		
	24	GTC Aux Rx Error portA An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.		
	23	GTC Update Complete portA A hardware initiated GTC update has completed with a sink attached to this port.		
	22	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
	21:20	Reserved		
	19:18	Reserved		
	17	GTC Aux Rx Error portD An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.		
	16	GTC Update Complete portD A hardware initiated GTC update has completed with a sink attached to this port.		
	15:10	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
	9	GTC Aux Rx Error portC An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.		
	8	GTC Update Complete portC A hardware initiated GTC update has completed with a sink attached to this port.		
	7:2	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ
	MBZ			
1	GTC Aux Rx Error portB An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.			
0	GTC Update Complete portB A hardware initiated GTC update has completed with a sink attached to this port.			

Half Precision Dual Source SIMD8 Message Data Payload Register

MDPR_DSH_SIMD8 - Half Precision Dual Source SIMD8 Message Data Payload Register				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	Src0 Data1 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 0 slot 1 data in this payload register		F16
		F16		
15:0	Src0 Data0 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 0 slot 0 data in this payload register		F16	
	F16			
1	31:16	Src0 Data3 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 0 slot 3 data in this payload register		F16
		F16		
15:0	Src0 Data2 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 0 slot 2 data in this payload register		F16	
	F16			
2	31:16	Src0 Data5 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 0 slot 5 data in this payload register		F16
		F16		
15:0	Src0 Data4 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 0 slot 4 data in this payload register		F16	
	F16			
3	31:16	Src0 Data7 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 0 slot 7 data in this payload register		F16
		F16		
15:0	Src0 Data6 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 0 slot 6 data in this payload register		F16	
	F16			
4	31:16	Src1 Data1 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 1 slot 1 data in this payload register		F16
		F16		
15:0	Src1 Data0 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">F16</td></tr></table> Specifies the source 1 slot 0 data in this payload register		F16	
	F16			

MDPR_DSH_SIMD8 - Half Precision Dual Source SIMD8 Message Data Payload Register				
5	31:16	<p>Src1 Data3</p> <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> <p>Specifies the source 1 slot 3 data in this payload register</p>	Format:	F16
	Format:	F16		
15:0	<p>Src1 Data2</p> <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> <p>Specifies the source 1 slot 2 data in this payload register</p>	Format:	F16	
Format:	F16			
6	31:16	<p>Src1 Data5</p> <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> <p>Specifies the source 1 slot 5 data in this payload register</p>	Format:	F16
	Format:	F16		
15:0	<p>Src1 Data4</p> <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> <p>Specifies the source 1 slot 4 data in this payload register</p>	Format:	F16	
Format:	F16			
7	31:16	<p>Src1 Data7</p> <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> <p>Specifies the source 1 slot 7 data in this payload register</p>	Format:	F16
	Format:	F16		
15:0	<p>Src1 Data6</p> <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> <p>Specifies the source 1 slot 6 data in this payload register</p>	Format:	F16	
Format:	F16			

Half Precision OM Replicated SIMD16 Render Target Data Payload

MDP_RTWH_M16REP - Half Precision OM Replicated SIMD16 Render Target Data Payload				
Source:	BSpec			
Size (in bits):	512			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	oMask <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_OMASK</td> </tr> </table> Slots [15:0] oMask	Format:	MDPR_OMASK
Format:	MDPR_OMASK			
1.0-1.7	255:0	RGBA <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_H_RGBA</td> </tr> </table> RGBA for all slots [15:0]	Format:	MDPR_H_RGBA
Format:	MDPR_H_RGBA			

Half Precision OM S0A SIMD8 Render Target Data Payload

MDP_RTWH_MA8 - Half Precision OM S0A SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha Format: MDPR_H_SIMD8 Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	oMask Format: MDPR_OMASK Slots [7:0] oMask. Upper half ignored.
2.0-2.7	255:0	Red Format: MDPR_H_SIMD8 Slots [7:0] Red
3.0-3.7	255:0	Green Format: MDPR_H_SIMD8 Slots [7:0] Green
4.0-4.7	255:0	Blue Format: MDPR_H_SIMD8 Slots [7:0] Blue
5.0-5.7	255:0	Alpha Format: MDPR_H_SIMD8 Slots [7:0] Alpha

Half Precision OM S0A SIMD16 Render Target Data Payload

MDP_RTWH_MA16 - Half Precision OM S0A SIMD16 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Source 0 Alpha
1.0-1.7	255:0	oMask Format: MDPR_OMASK Slots [15:0] oMask
2.0-2.7	255:0	Red[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Red
3.0-3.7	255:0	Green[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Green
4.0-4.7	255:0	Blue[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Blue
5.0-5.7	255:0	Alpha[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Alpha

Half Precision OM SIMD8 Render Target Data Payload

MDP_RTWH_M8 - Half Precision OM SIMD8 Render Target Data Payload				
Source:	BSpec			
Size (in bits):	1280			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	oMask <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_OMASK</td> </tr> </table> Slots [7:0] oMask. Upper half ignored.	Format:	MDPR_OMASK
Format:	MDPR_OMASK			
1.0-1.7	255:0	Red <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_H_SIMD8</td> </tr> </table> Slots [7:0] Red	Format:	MDPR_H_SIMD8
Format:	MDPR_H_SIMD8			
2.0-2.7	255:0	Green <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_H_SIMD8</td> </tr> </table> Slots [7:0] Green	Format:	MDPR_H_SIMD8
Format:	MDPR_H_SIMD8			
3.0-3.7	255:0	Blue <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_H_SIMD8</td> </tr> </table> Slots [7:0] Blue	Format:	MDPR_H_SIMD8
Format:	MDPR_H_SIMD8			
4.0-4.7	255:0	Alpha <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_H_SIMD8</td> </tr> </table> Slots [7:0] Alpha	Format:	MDPR_H_SIMD8
Format:	MDPR_H_SIMD8			

MDP_RTWH_SMA8 - Half Precision OS OM S0A SIMD8 Render Target Data Payload

6.0-6.7	255:0	Stencil	
		Format:	MDPR_STENCIL
		Slots [7:0] Stencil	

Half Precision OS S0A SIMD8 Render Target Data Payload

MDP_RTWH_SA8 - Half Precision OS S0A SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha Format: MDPR_H_SIMD8 Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	Red Format: MDPR_H_SIMD8 Slots [7:0] Red
2.0-2.7	255:0	Green Format: MDPR_H_SIMD8 Slots [7:0] Green
3.0-3.7	255:0	Blue Format: MDPR_H_SIMD8 Slots [7:0] Blue
4.0-4.7	255:0	Alpha Format: MDPR_H_SIMD8 Slots [7:0] Alpha
5.0-5.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] Stencil

Half Precision OS SIMD8 Render Target Data Payload

MDP_RTWH_S8 - Half Precision OS SIMD8 Render Target Data Payload				
Source:	BSpec			
Size (in bits):	1280			
Default Value:	0x00000000, 0x00000000,			
DWord	Bit	Description		
0.0-0.7	255:0	Red <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_H_SIMD8</td> </tr> </table> Slots [7:0] Red	Format:	MDPR_H_SIMD8
Format:	MDPR_H_SIMD8			
1.0-1.7	255:0	Green <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_H_SIMD8</td> </tr> </table> Slots [7:0] Green	Format:	MDPR_H_SIMD8
Format:	MDPR_H_SIMD8			
2.0-2.7	255:0	Blue <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_H_SIMD8</td> </tr> </table> Slots [7:0] Blue	Format:	MDPR_H_SIMD8
Format:	MDPR_H_SIMD8			
3.0-3.7	255:0	Alpha <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_H_SIMD8</td> </tr> </table> Slots [7:0] Alpha	Format:	MDPR_H_SIMD8
Format:	MDPR_H_SIMD8			
4.0-4.7	255:0	Stencil <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDPR_STENCIL</td> </tr> </table> Slots [7:0] Stencil	Format:	MDPR_STENCIL
Format:	MDPR_STENCIL			

MDP_RTWH_SZMA8 - Half Precision OS SZ OM S0A SIMD8 Render Target Data Payload

7.0-7.7	255:0	Stencil	
		Format:	MDPR_STENCIL
		Slots [7:0] Stencil	

MDP_RTWH_SZM8DS - Half Precision OS SZ OM SIMD8 Dual Source Render Target Data Payload		
6.0-6.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] or [15:8] of Stencil

MDP_RTWH_SZM8 - Half Precision OS SZ OM SIMD8 Render Target Data Payload		
6.0-6.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] Stencil

Half Precision OS SZ S0A SIMD8 Render Target Data Payload

MDP_RTWH_SZA8 - Half Precision OS SZ S0A SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1792	
Default Value:	0x00000000, 0x00000000,	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha Format: MDPR_H_SIMD8 Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	Red Format: MDPR_H_SIMD8 Slots [7:0] Red
2.0-2.7	255:0	Green Format: MDPR_H_SIMD8 Slots [7:0] Green
3.0-3.7	255:0	Blue Format: MDPR_H_SIMD8 Slots [7:0] Blue
4.0-4.7	255:0	Alpha Format: MDPR_H_SIMD8 Slots [7:0] Alpha
5.0-5.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] Source Depth

MDP_RTWH_SZA8 - Half Precision OS SZ S0A SIMD8 Render Target Data Payload		
6.0-6.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] Stencil

Half Precision OS SZ SIMD8 Render Target Data Payload

MDP_RTWH_SZ8 - Half Precision OS SZ SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Red Format: MDPR_H_SIMD8 Slots [7:0] Red
1.0-1.7	255:0	Green Format: MDPR_H_SIMD8 Slots [7:0] Green
2.0-2.7	255:0	Blue Format: MDPR_H_SIMD8 Slots [7:0] Blue
3.0-3.7	255:0	Alpha Format: MDPR_H_SIMD8 Slots [7:0] Alpha
4.0-4.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] Source Depth
5.0-5.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] Stencil

Half Precision Replicated Pixel Render Target Data Payload Register

MDPR_H_RGBA - Half Precision Replicated Pixel Render Target Data Payload Register				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	Green <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U16</td> </tr> </table> Specifies the value of all slots' green channel.	Format:	U16
	Format:	U16		
15:0	Red <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U16</td> </tr> </table> Specifies the value of all slots' red channel.	Format:	U16	
Format:	U16			
1	31:16	Alpha <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U16</td> </tr> </table> Specifies the value of all slots' alpha channel.	Format:	U16
	Format:	U16		
15:0	Blue <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U16</td> </tr> </table> Specifies the value of all slots' blue channel.	Format:	U16	
Format:	U16			
2-7	191:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">Ignore</td> </tr> </table> Ignored	Format:	Ignore
Format:	Ignore			

Half Precision Replicated SIMD16 Render Target Data Payload

MDP_RTWH_16REP - Half Precision Replicated SIMD16 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	RGBA Format: MDPR_H_RGBA RGBA for all slots [15:0]

Half Precision S0A SIMD8 Render Target Data Payload

MDP_RTWH_A8 - Half Precision S0A SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha Format: MDPR_H_SIMD8 Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	Red Format: MDPR_H_SIMD8 Slots [7:0] Red
2.0-2.7	255:0	Green Format: MDPR_H_SIMD8 Slots [7:0] Green
3.0-3.7	255:0	Blue Format: MDPR_H_SIMD8 Slots [7:0] Blue
4.0-4.7	255:0	Alpha Format: MDPR_H_SIMD8 Slots [7:0] Alpha

Half Precision S0A SIMD16 Render Target Data Payload

MDP_RTWH_A16 - Half Precision S0A SIMD16 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Source 0 Alpha
1.0-1.7	255:0	Red[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Red
2.0-2.7	255:0	Green[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Green
3.0-3.7	255:0	Blue[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Blue
4.0-4.7	255:0	Alpha[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Alpha

Half Precision SIMD8 Message Data Payload Register

MDPR_H_SIMD8 - Half Precision SIMD8 Message Data Payload Register				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	Data1 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">F16</td> </tr> </table> Specifies the slot 1 data in this payload register	Format:	F16
	Format:	F16		
15:0	Data0 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">F16</td> </tr> </table> Specifies the slot 0 data in this payload register	Format:	F16	
Format:	F16			
1	31:16	Data3 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">F16</td> </tr> </table> Specifies the slot 3 data in this payload register	Format:	F16
	Format:	F16		
15:0	Data2 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">F16</td> </tr> </table> Specifies the slot 2 data in this payload register	Format:	F16	
Format:	F16			
2	31:16	Data5 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">F16</td> </tr> </table> Specifies the slot 5 data in this payload register	Format:	F16
	Format:	F16		
15:0	Data4 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">F16</td> </tr> </table> Specifies the slot 4 data in this payload register	Format:	F16	
Format:	F16			
3	31:16	Data7 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">F16</td> </tr> </table> Specifies the slot 7 data in this payload register	Format:	F16
	Format:	F16		
15:0	Data6 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">F16</td> </tr> </table> Specifies the slot 6 data in this payload register	Format:	F16	
Format:	F16			

MDPR_H_SIMD8 - Half Precision SIMD8 Message Data Payload Register

4-7	127:0	Reserved	
		Format:	Ignore
		Ignored	

Half Precision SIMD8 Render Target Data Payload

MDP_RTWH_8 - Half Precision SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1024	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Red Format: MDPR_H_SIMD8 Slots [7:0] Red
1.0-1.7	255:0	Green Format: MDPR_H_SIMD8 Slots [7:0] Green
2.0-2.7	255:0	Blue Format: MDPR_H_SIMD8 Slots [7:0] Blue
3.0-3.7	255:0	Alpha Format: MDPR_H_SIMD8 Slots [7:0] Alpha

Half Precision SIMD16 Message Data Payload Register

MDPR_H_SIMD16 - Half Precision SIMD16 Message Data Payload Register				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	Data1 <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> Specifies the slot 1 data in this payload register	Format:	F16
	Format:	F16		
15:0	Data0 <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> Specifies the slot 0 data in this payload register	Format:	F16	
Format:	F16			
1	31:16	Data3 <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> Specifies the slot 3 data in this payload register	Format:	F16
	Format:	F16		
15:0	Data2 <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> Specifies the slot 2 data in this payload register	Format:	F16	
Format:	F16			
2	31:16	Data5 <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> Specifies the slot 5 data in this payload register	Format:	F16
	Format:	F16		
15:0	Data4 <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> Specifies the slot 4 data in this payload register	Format:	F16	
Format:	F16			
3	31:16	Data7 <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> Specifies the slot 7 data in this payload register	Format:	F16
	Format:	F16		
15:0	Data6 <table border="1"> <tr> <td>Format:</td> <td>F16</td> </tr> </table> Specifies the slot 6 data in this payload register	Format:	F16	
Format:	F16			

MDPR_H_SIMD16 - Half Precision SIMD16 Message Data Payload Register				
4	31:16	<p>Data9</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">F16</td> </tr> </table> <p>Specifies the slot 9 data in this payload register</p>	Format:	F16
	Format:	F16		
15:0	<p>Data8</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">F16</td> </tr> </table> <p>Specifies the slot 8 data in this payload register</p>	Format:	F16	
Format:	F16			
5	31:16	<p>Data11</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">F16</td> </tr> </table> <p>Specifies the slot 11 data in this payload register</p>	Format:	F16
	Format:	F16		
15:0	<p>Data10</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">F16</td> </tr> </table> <p>Specifies the slot 10 data in this payload register</p>	Format:	F16	
Format:	F16			
6	31:16	<p>Data13</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">F16</td> </tr> </table> <p>Specifies the slot 13 data in this payload register</p>	Format:	F16
	Format:	F16		
15:0	<p>Data12</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">F16</td> </tr> </table> <p>Specifies the slot 12 data in this payload register</p>	Format:	F16	
Format:	F16			
7	31:16	<p>Data15</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">F16</td> </tr> </table> <p>Specifies the slot 15 data in this payload register</p>	Format:	F16
	Format:	F16		
15:0	<p>Data14</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">F16</td> </tr> </table> <p>Specifies the slot 14 data in this payload register</p>	Format:	F16	
Format:	F16			



MDP_RTWH_ZMA8 - Half Precision SZ OM S0A SIMD8 Render Target Data Payload				
6.0-6.7	255:0	Source Depth <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Source Depth	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			



MDP_RTWH_ZMA16 - Half Precision SZ OM S0A SIMD16 Render Target Data Payload		
6.0-6.7	255:0	Source Depth[7:0]
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Source Depth
Format:	MDP_DW_SIMD8	
7.0-7.7	255:0	Source Depth[15:8]
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Source Depth
Format:	MDP_DW_SIMD8	

Half Precision SZ OM SIMD8 Render Target Data Payload

MDP_RTWH_ZM8 - Half Precision SZ OM SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	oMask Format: MDPR_OMASK Slots [7:0] oMask. Upper half ignored.
1.0-1.7	255:0	Red Format: MDPR_H_SIMD8 Slots [7:0] Red
2.0-2.7	255:0	Green Format: MDPR_H_SIMD8 Slots [7:0] Green
3.0-3.7	255:0	Blue Format: MDPR_H_SIMD8 Slots [7:0] Blue
4.0-4.7	255:0	Alpha Format: MDPR_H_SIMD8 Slots [7:0] Alpha
5.0-5.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] Source Depth



MDP_RTWH_ZM16 - Half Precision SZ OM SIMD16 Render Target Data Payload				
6.0-6.7	255:0	Source Depth[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Source Depth	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			

Half Precision SZ S0A SIMD8 Render Target Data Payload

MDP_RTWH_ZA8 - Half Precision SZ S0A SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha Format: MDPR_H_SIMD8 Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	Red Format: MDPR_H_SIMD8 Slots [7:0] Red
2.0-2.7	255:0	Green Format: MDPR_H_SIMD8 Slots [7:0] Green
3.0-3.7	255:0	Blue Format: MDPR_H_SIMD8 Slots [7:0] Blue
4.0-4.7	255:0	Alpha Format: MDPR_H_SIMD8 Slots [7:0] Alpha
5.0-5.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] Source Depth

MDP_RTWH_ZA16 - Half Precision SZ S0A SIMD16 Render Target Data Payload		
6.0-6.7	255:0	Source Depth[15:8] Format: MDP_DW_SIMD8 Slots [15:8] Source Depth

Half Precision SZ SIMD8 Render Target Data Payload

MDP_RTWH_Z8 - Half Precision SZ SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Red Format: MDPR_H_SIMD8 Slots [7:0] Red
1.0-1.7	255:0	Green Format: MDPR_H_SIMD8 Slots [7:0] Green
2.0-2.7	255:0	Blue Format: MDPR_H_SIMD8 Slots [7:0] Blue
3.0-3.7	255:0	Alpha Format: MDPR_H_SIMD8 Slots [7:0] Alpha
4.0-4.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] Source Depth

Half Precision SZ SIMD16 Render Target Data Payload

MDP_RTWH_Z16 - Half Precision SZ SIMD16 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Red[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Red
1.0-1.7	255:0	Green[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Green
2.0-2.7	255:0	Blue[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Blue
3.0-3.7	255:0	Alpha[15:0] Format: MDPR_H_SIMD16 Slots [15:0] Alpha
4.0-4.7	255:0	Source Depth[7:0] Format: MDP_DW_SIMD8 Slots [7:0] Source Depth
5.0-5.7	255:0	Source Depth[15:8] Format: MDP_DW_SIMD8 Slots [15:8] Source Depth

Hardware-Detected Error Bit Definitions

Hardware-Detected Error Bit Definitions							
Source:	RenderCS						
Size (in bits):	32						
Default Value:	0x00000000						
DWord	Bit	Description					
0	31:3	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ			
		MBZ					
	2	Command Privilege Violation Error This bit is set if a command classified as privileged is parsed in a non-privileged batch buffer. The command will be converted to a NOOP and parsing will continue.					
	1	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ			
	MBZ						
0	Instruction Error This bit is set when the Renderer Instruction Parser detects an error while parsing an instruction. Instruction errors include: <ul style="list-style-type: none"> Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported). Defeatured MI Instruction Opcodes: <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>1</td> <td></td> <td>Instruction Error detected</td> </tr> </tbody> </table>	Value	Name	Description	1		Instruction Error detected
Value	Name	Description					
1		Instruction Error detected					
Programming Notes							
This error indications cannot be cleared except by reset (i.e., it is a fatal error).							

Hardware Status Page Layout		
		The contents of the Ring Buffer Head Pointer register (register DWord 1) are written to this location either as result of an MI_REPORT_HEAD instruction or as the result of an "automatic report" (see RINGBUF registers).
5..15	31:0	Reserved Must not be used.
16..27	31:0	Context Status DWords
28..30	31:0	Reserved Must not be used.
31	31:0	Last Written Status Offset
32..39	31:0	Reserved
40..46	31:0	Reserved
47	31:0	Reserved
48..1023	31:0	General Purpose These locations can be used for general purpose via the MI_STORE_DATA_INDEX or MI_STORE_DATA_IMM instructions.

HCP_PAK_INSERT_OBJECT_INDIRECT_PAYLOAD

HCP_PAK_INSERT_OBJECT_INDIRECT_PAYLOAD				
Source:	VideoCS			
Size (in bits):	128			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	Indirect Payload Data Size in bits Format: U32 Number of bits to be inserted. Not including those skipped bytes in the beginning.		
1..2	63:0	Indirect Payload Base Address Format: SplitBaseAddress64ByteAligned 48-bit address of the indirect payload data in memory buffer. <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Programming Notes</td> </tr> <tr> <td>Payload must begin in a byte position, but the payload can be ended in a bit position.</td> </tr> </table>	Programming Notes	Payload must begin in a byte position, but the payload can be ended in a bit position.
Programming Notes				
Payload must begin in a byte position, but the payload can be ended in a bit position.				
3	31:0	Indirect Payload Base Address Format: MemoryAddressAttributes		

Header Forbidden Message Descriptor Control Field

MDC_MHF - Header Forbidden Message Descriptor Control Field															
Source:	BSpec														
Size (in bits):	1														
Default Value:	0x00000000														
DWord	Bit	Description													
0	0	<p>Message Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>Enumeration</td> </tr> <tr> <td colspan="2">Indicates the message forbids a message header.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>0h</td> <td>No [Default]</td> <td>Message header is not present</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td>Not used</td> </tr> </table>	Format:	Enumeration	Indicates the message forbids a message header.		Value	Name	Description	0h	No [Default]	Message header is not present	1h	Reserved	Not used
Format:	Enumeration														
Indicates the message forbids a message header.															
Value	Name	Description													
0h	No [Default]	Message header is not present													
1h	Reserved	Not used													

Header Present Message Descriptor Control Field

MDC_MHP - Header Present Message Descriptor Control Field													
Source:	BSpec												
Size (in bits):	1												
Default Value:	0x00000000												
DWord	Bit	Description											
0	0	<p>Message Header Present</p> <table border="1"> <tr> <td>Format:</td> <td>Enumeration</td> </tr> </table> <p>Specifies if the message uses the optional message header.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No</td> <td>Message header is not present</td> </tr> <tr> <td>1h</td> <td>Yes</td> <td>Message header is present</td> </tr> </tbody> </table>	Format:	Enumeration	Value	Name	Description	0h	No	Message header is not present	1h	Yes	Message header is present
Format:	Enumeration												
Value	Name	Description											
0h	No	Message header is not present											
1h	Yes	Message header is present											

Header Required Message Descriptor Control Field

MDC_MHR - Header Required Message Descriptor Control Field																	
Source:	BSpec																
Size (in bits):	1																
Default Value:	0x00000001																
DWord	Bit	Description															
0	0	<p>Message Header Present</p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">Enumeration</td> </tr> <tr> <td colspan="3">Indicates the message requires a message header.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>0h</td> <td>Reserved</td> <td>Not used</td> </tr> <tr> <td>1h</td> <td>Yes [Default]</td> <td>Message header is present</td> </tr> </table>	Format:	Enumeration		Indicates the message requires a message header.			Value	Name	Description	0h	Reserved	Not used	1h	Yes [Default]	Message header is present
Format:	Enumeration																
Indicates the message requires a message header.																	
Value	Name	Description															
0h	Reserved	Not used															
1h	Yes [Default]	Message header is present															

HW Generated BINDING_TABLE_STATE

HW Generated BINDING_TABLE_STATE				
Source:	BSpec			
Size (in bits):	16			
Default Value:	0x00000000			
Description				
<p>The binding table binds surfaces to logical resource indices used by shaders and other compute engine kernels. The HW generated Binding_Table_State have different format than the SW generated Binding_Table_State. The HW generated Binding_Table_State is stored as an array of 256 elements, each of which contains one word as defined here. The start of each element is spaced one word apart. The first element of the binding table is aligned to a 64-byte boundary. Binding table indexes beyond 256 will automatically be mapped to entry 0 by the HW, w/ the exception of any messages which support the special indexes 240 through 255, inclusive.</p>				
DWord	Bit	Description		
0	15:0	Surface State Pointer <table border="1"> <tr> <td>Format:</td> <td>SurfaceStateOffset[21:6]</td> </tr> </table>	Format:	SurfaceStateOffset[21:6]
Format:	SurfaceStateOffset[21:6]			

Hword 1 Block Data Payload

MDP_HW1 - Hword 1 Block Data Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	Hword <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U256</td> </tr> </table> Specifies the Hword data	Format:	U256
Format:	U256			

Hword 2 Block Data Payload

MDP_HW2 - Hword 2 Block Data Payload				
Source:	BSpec			
Size (in bits):	512			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<p>Hword0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U256</td> </tr> </table> <p>Specifies the Hword data for element 0</p>	Format:	U256
Format:	U256			
1.0-1.7	255:0	<p>Hword1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U256</td> </tr> </table> <p>Specifies the Hword data for element 1</p>	Format:	U256
Format:	U256			

MDP_HW8 - Hword 8 Block Data Payload		
6.0-6.7	255:0	Hword6 Format: U256 Specifies the Hword data for element 6
		Hword7 Format: U256 Specifies the Hword data for element 7

Hword Channel Mode Message Header Control

MHC_A64_CMODE - Hword Channel Mode Message Header Control		
Source:	BSpec	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31	Reserved 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.
	30:0	Reserved Format: Ignore Ignored

Hword Register Blocks Message Descriptor Control Field

MDC_DB_HW - Hword Register Blocks Message Descriptor Control Field																					
Source:	BSpec																				
Size (in bits):	2																				
Default Value:	0x00000000																				
DWord	Bit	Description																			
0	1:0	<p>Register Blocks</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> <tr> <td colspan="2">Specifies the number of Hword blocks to be read or written</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">HW1</td> <td style="text-align: center;">1 Hword register</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">HW2</td> <td style="text-align: center;">2 Hword registers</td> </tr> <tr> <td style="text-align: center;">02h</td> <td style="text-align: center;">HW4</td> <td style="text-align: center;">4 Hword registers</td> </tr> <tr> <td style="text-align: center;">03h</td> <td style="text-align: center;">HW8</td> <td style="text-align: center;">8 Hword registers</td> </tr> </table>	Format:	Enumeration	Specifies the number of Hword blocks to be read or written		Value	Name	Description	00h	HW1	1 Hword register	01h	HW2	2 Hword registers	02h	HW4	4 Hword registers	03h	HW8	8 Hword registers
Format:	Enumeration																				
Specifies the number of Hword blocks to be read or written																					
Value	Name	Description																			
00h	HW1	1 Hword register																			
01h	HW2	2 Hword registers																			
02h	HW4	4 Hword registers																			
03h	HW8	8 Hword registers																			

Ignored Message Header

MH_IGNORE - Ignored Message Header				
Source:	DataPort 0			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
Some messages require a message header or have an optional message header, but do not use any information in the header.				
DWord	Bit	Description		
0-7	255:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

Inline Data Description for MFD_AVC_BSD_Object

Inline Data Description for MFD_AVC_BSD_Object															
Source:	VideoCS														
Size (in bits):	96														
Default Value:	0x00000000, 0x00000000, 0x00000000														
This structure includes all the required Slice Header parameters and error handling settings for AVC_BSD_OBJECT Command (DW3..DW5).															
DWord	Bit	Description													
0	31	Concealment Method This field specifies the method used for concealment when error is detected. If set, a copy from collocated macroblock location is performed from the concealment reference indicated by the ConCeal_Pic_Id field. If it is not set, a copy from the current picture is performed using Intra 16x16 Prediction method.													
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Intra 16x16 Prediction</td> </tr> <tr> <td>1</td> <td></td> <td>Inter P Copy</td> </tr> </tbody> </table>	Value	Name	Description	0		Intra 16x16 Prediction	1		Inter P Copy				
		Value	Name	Description											
		0		Intra 16x16 Prediction											
	1		Inter P Copy												
	30	Init Current MB Number When set, the current Slice_Start_MB_Num, Slice_MB_Start_Hor_Pos and Slice_MB_Start_Vert_Pos fields will be used to initialize the Current_MB_Number register. This effectively disables the concealment capability.													
	29	Intra PredMode (4x4/8x8 Luma) Error Control Bit This field controls if AVC decoder will fix Intra Prediction Mode if the decoded value is incorrect according to MB position													
	28:27	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>AVC decoder will detect and fix IntraPredMode (4x4/8x8 Luma) Errors.</td> </tr> <tr> <td>1</td> <td></td> <td>AVC decoder will NOT detect IntraPredMode (4x4/8x8 Luma) Errors. The wrong IntraPredMode value will be retained.</td> </tr> </tbody> </table>	Value	Name	Description	0		AVC decoder will detect and fix IntraPredMode (4x4/8x8 Luma) Errors.	1		AVC decoder will NOT detect IntraPredMode (4x4/8x8 Luma) Errors. The wrong IntraPredMode value will be retained.	MB Error Concealment B Temporal Prediction mode These two bits control how the reference L0/L1 are overridden in B temporal slice.			
			Value	Name	Description										
			0		AVC decoder will detect and fix IntraPredMode (4x4/8x8 Luma) Errors.										
1				AVC decoder will NOT detect IntraPredMode (4x4/8x8 Luma) Errors. The wrong IntraPredMode value will be retained.											
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>[Default]</td> <td>Both Reference Indexes L0/L1 are forced to 0 during Concealment</td> </tr> <tr> <td>01b</td> <td></td> <td>Only Reference Index L1 is forced to 0; Reference Index L0 is forced to -1</td> </tr> <tr> <td>10b</td> <td></td> <td>Only Reference Index L0 is forced to 0; Reference Index L1 is forced to -1</td> </tr> <tr> <td>11b</td> <td>Reserved</td> <td>Invalid</td> </tr> </tbody> </table>	Value		Name	Description	00b	[Default]	Both Reference Indexes L0/L1 are forced to 0 during Concealment	01b		Only Reference Index L1 is forced to 0; Reference Index L0 is forced to -1	10b		Only Reference Index L0 is forced to 0; Reference Index L1 is forced to -1	11b	Reserved
Value	Name	Description													
00b	[Default]	Both Reference Indexes L0/L1 are forced to 0 during Concealment													
01b		Only Reference Index L1 is forced to 0; Reference Index L0 is forced to -1													
10b		Only Reference Index L0 is forced to 0; Reference Index L1 is forced to -1													
11b	Reserved	Invalid													
26	Reserved Format:	MBZ													

Inline Data Description for MFD_AVC_BSD_Object		
25	MB Error Concealment B Temporal Motion Vectors Override Enable Flag	
	During MB Error Concealment on B slice with Temporal Direct Prediction, motion vectors are forced to 0 to improve image quality. This bit can be set to preserve the original weight prediction.	
	Value	Name Description
	0	[Default] Predicted Motion Vectors are used during MB Concealment
1	Motion Vectors are Overridden to 0 during MB Concealment	
24	MB Error Concealment B Temporal Weight Prediction Disable Flag	
	During MB Error Concealment on B slice with Temporal Direct Prediction, weight prediction is disabled to improve image quality. This bit can be set to preserve the original weight prediction.	
	Value	Name Description
	0	[Default] Weight Prediction is Disabled during MB Concealment
1	Weight Prediction will not be overridden during MB Concealment	
23:22	Reserved	
	Format:	MBZ
21:16	Concealment Picture ID	
	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.	
	Bit Filed	Value Defenition
	21	0 Frame Picture
	21	1 Field picture
20:16	All Frame Store Index[4:0]	
15	Reserved	
	Format:	MBZ
14	BSD Premature Complete Error Handling	
	BSD Premature Complete Error occurs in situation where the Slice decode is completed but there are still data in the bitstream.	
	Value	Name Description
	1	Set the interrupt to the driver (provide MMIO registers for MB address R/W)
0	Ignore the error and continue (masked the interrupt), assume the hardware automatically performs the error handling	
13	Reserved	
	Format:	MBZ

Inline Data Description for MFD_AVC_BSD_Object

12	MPR Error (MV out of range) Handling Software must follow the action for each Value as follow:	
	Value	Name
	Description	
	1	Set the interrupt to the driver (provide MMIO registers for MB address R/W)
	0	Ignore the error and continue (masked the interrupt), assume the hardware automatically performs the error handling
11	Reserved Format: _____ MBZ	
10	Entropy Error Handling Software must follow the action for each Value as follow:	
	Value	Name
	Description	
	1	Set the interrupt to the driver (provide MMIO registers for MB address R/W).
	0	Ignore the error and continue (masked the interrupt), assume the hardware automatically perform the error handling.
9	Reserved Format: _____ MBZ	
8	MB Header Error Handling Software must follow the action for each Value as follow:	
	Value	Name
	Description	
	1	Set the interrupt to the driver (provide MMIO registers for MB address R/W).
	0	Ignore the error and continue (masked the interrupt), assume the hardware automatically perform the error concealment.
7:6	MB Error Concealment B Spatial Prediction mode These two bits control how the reference L0/L1 are overridden in B spatial slice.	
	Value	Name
	Description	
	00b	[Default] Both Reference Indexes L0/L1 are forced to 0 during Concealment
	01b	Only Reference Index L1 is forced to 0; Reference Index L0 is forced to -1
	10b	Only Reference Index L0 is forced to 0; Reference Index L1 is forced to -1
	11b	Reserved Invalid
5	Reserved Format: _____ MBZ	
4	MB Error Concealment B Spatial Motion Vectors Override Disable Flag During MB Error Concealment on B slice with Spatial Direct Prediction, motion vectors are forced to 0 to improve image quality. This bit can be set to use the predicted motion vectors instead. This bit does not affect normal decoded MB.	
	Value	Name
	Description	
	0	[Default] Motion Vectors are Overridden to 0 during MB Concealment
	1	Predicted Motion Vectors are used during MB Concealment

Inline Data Description for MFD_AVC_BSD_Object											
1	3	<p>MB Error Concealment B Spatial Weight Prediction Disable Flag</p> <p>During MB Error Concealment on B slice with Spatial Direct Prediction, weight prediction is disabled to improve image quality. This bit can be set to preserve the original weight prediction. This bit does not affect normal decoded MB.</p> <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 style="text-align: center;">0</td> <td style="text-align: center;">[Default]</td> <td>Weight Prediction is Disabled during MB Concealment.</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>Weight Prediction will not be overridden during MB Concealment.</td> </tr> </tbody> </table>	Value	Name	Description	0	[Default]	Weight Prediction is Disabled during MB Concealment.	1		Weight Prediction will not be overridden during MB Concealment.
	Value	Name	Description								
	0	[Default]	Weight Prediction is Disabled during MB Concealment.								
	1		Weight Prediction will not be overridden during MB Concealment.								
	2	<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									
	1	<p>MB Error Concealment P Slice Motion Vectors Override Disable Flag</p> <p>During MB Error Concealment on P slice, motion vectors are forced to 0 to improve image quality. This bit can be set to use the predicted motion vectors instead. This bit does not affect normal decoded MB.</p> <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 style="text-align: center;">0</td> <td style="text-align: center;">[Default]</td> <td>Motion Vectors are Overridden to 0 during MB Concealment</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>Predicted Motion Vectors are used during MB Concealment</td> </tr> </tbody> </table>	Value	Name	Description	0	[Default]	Motion Vectors are Overridden to 0 during MB Concealment	1		Predicted Motion Vectors are used during MB Concealment
	Value	Name	Description								
	0	[Default]	Motion Vectors are Overridden to 0 during MB Concealment								
	1		Predicted Motion Vectors are used during MB Concealment								
	0	<p>MB Error Concealment P Slice Weight Prediction Disable Flag</p> <p>During MB Error Concealment on P slice, weight prediction is disabled to improve image quality. This bit can be set to preserve the original weight prediction. This bit does not affect normal decoded MB.</p> <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 style="text-align: center;">0</td> <td style="text-align: center;">[Default]</td> <td>Weight Prediction is Disabled during MB Concealment.</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>Weight Prediction will not be overridden during MB Concealment.</td> </tr> </tbody> </table>	Value	Name	Description	0	[Default]	Weight Prediction is Disabled during MB Concealment.	1		Weight Prediction will not be overridden during MB Concealment.
	Value	Name	Description								
0	[Default]	Weight Prediction is Disabled during MB Concealment.									
1		Weight Prediction will not be overridden during MB Concealment.									
31:16	<p>First MB Byte Offset of Slice Data or Slice Header</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> <tr> <td>MFX supports only DXVA2 Long and Short Format.</td> </tr> </table>	Programming Notes	MFX supports only DXVA2 Long and Short Format.								
Programming Notes											
MFX supports only DXVA2 Long and Short Format.											
15:8	<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										
7	<p>Fix Prev Mb Skipped</p> <p>Enables an alternative method for decoding mb_skipped, to cope with an encoder that codes a skipped MB as a direct MB with no coefficient.</p>										
6:5	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> <tr> <td>Please note that the field MUST be set to '0' at this time.</td> </tr> </table>	Format:	MBZ	Programming Notes	Please note that the field MUST be set to '0' at this time.						
Format:	MBZ										
Programming Notes											
Please note that the field MUST be set to '0' at this time.											

Inline Data Description for MFD_AVC_BSD_Object			
4	Emulation Prevention Byte Present		
	Value	Name	
	0	H/W needs to perform Emulation Byte Removal	
	1	H/W does not need to perform Emulation Byte Removal	
3	LastSlice Flag		
	It is needed for both error concealment at the end of a picture (so, no more phantom slice). It is also needed to know to set the last MB in a picture correctly.		
	Value	Name	
	1	If the current Slice to be decoded is the very last slice of the current picture.	
2:0	First Macroblock (MB)Bit Offset		
	Exists If:	//AVC Long Format Only	
	Format:	U3	
	This field provides the bit offset of the first macroblock of the Slice in the first byte of the input compressed bitstream.		
2	31	I Slice Concealment Mode	
		This field controls how AVC decoder handle MB concealment in I Slice	
		Value	Name
	0	Intra Concealment	
	1	Inter Concealment	
	Programming Notes		
	If this field is set to "1" (Inter Concealment), driver must provide a valid reference picture (programmed using "Concealment Reference Picture" field) for concealment reference picture. In this mode, weight prediction is disabled and motion vectors are forced to 0 as well.		
	30	Reserved	
		Format:	MBZ
	29:24	Concealment Reference Picture + Field Bit	
Format:		U6	
This field provides the concealment reference picture for hardware to conceal in case driver wants to specify one concealment picture. This field matches with the DPB order sent to hardware. This field applies to all I/P/B slices			
Bit Filed		Value	
29		MBZ	
28:25		All	
24	All		
Defenition			
is reserved for future expansion			
Reference PICTURE Number			
Field Bit(if the current picture is a field picture [Frame picture must be 0])			

Inline Data Description for MFD_AVC_BSD_Object			
23	P Slice Concealment Mode		
	This field controls how AVC decoder handle MB concealment in P Slice		
	Value	Name	
	1	Intra Concealment	
	0	Inter Concealment	
22:19	Reserved		
	Format:	MBZ	
18:16	P Slice Inter Concealment Mode		
	This field controls how AVC decoder select reference picture for Concealment in P Slice.		
	Value	Name	Description
	000b		Top of Reference List L0 (Use top entry of Reference List L0)
	001b		Driver Specified Concealment Reference
	010b		Predicted Reference (Use reference picture predicted using P-Skip Algorithm)
	011b		Temporal Closest (Using POC to select the closest forward picture) [For L0: Closest POC smaller than current POC]
	100b		First Long Term Picture in Reference List L0 (If no long term picture available, use Temporal Closest Picture)
101b-111b	Reserved		
15	B Slice Concealment Mode		
	This field controls how AVC decoder handle MB concealment in B Slice		
	Value	Name	
	1	Intra Concealment	
	0	Inter Concealment	
14	Reserved		
	Format:	MBZ	
13:12	B Slice Inter Direct Type Concealment Mode		
	AVC decoder can use Spatial or Temporal Direct for B Skip/Direct. This field determine can override the mode on how AVC decoder handles MB concealment in B slice.		
	Value	Name	Description
	00b		Use Default Direct Type (slice programmed direct type)
	01b		Forced to Spatial Direct Only
	10b		Forced to Temporal Direct Only
11b		Spatial Direct without Temporal Component (MovingBlock information)	
11	Reserved		
	Format:	MBZ	

Inline Data Description for MFD_AVC_BSD_Object

10:8	B Slice Spatial Inter Concealment Mode	This field controls how AVC decoder select reference picture for Spatial Inter Concealment in B Slice.
	Value	Name
	Description	
	000b	Top of Reference List L0/L1 (Use top entry of Reference List L0/L1).
	001b	Driver Specified Concealment Reference
	011b	Temporal Closest (Using POC to select the closest forward picture) [For L0: Closest POC smaller than current POC] [For L1: Closest POC larger than current POC]
	100b	" First Long Term Picture in Reference List L0/L1 (If no long term picture available, use Temporal Closest Picture)
	101b-111b	Reserved
7	Reserved	
	Format:	MBZ
6:4	B Slice Temporal Inter Concealment Mode	This field controls how AVC decoder select reference picture for Temporal Inter Concealment in B Slice
	Value	Name
	Description	
	000b	Top of Reference List L0/L1 (Use top entry of Reference List L0/L1)
	001b	Driver Specified Concealment Reference
	010b	Predicted Reference (Use reference picture predicted using B-Skip Algorithm)
	011b	" Temporal Closest (Using POC to select the closest forward picture) [For L0: Closest POC smaller than current POC] [For L1: Closest POC larger than current POC]
	100b	First Long Term Picture in Reference List L0/L1 (If no long term picture available, use Temporal Closest Picture)
	101b-111b	Reserved
3:2	Reserved	
	Format:	MBZ

Inline Data Description for MFD_AVC_BSD_Object											
	1	<p>Intra 8x8/4x4 Prediction Error Concealment Control Bit</p> <p>This field controls if AVC goes into MB concealment mode (next MB) when an error is detected on Intra8x8/4x4 Prediction Mode (these 2 modes have fixed coding so it may not affect the 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>AVC decoder will NOT go into MB concealment when Intra8x8/4x4 Prediction mode is incorrect.</td> </tr> <tr> <td>1</td> <td></td> <td>AVC decoder will go into MB concealment when Intra8x8/4x4 Prediction mode is incorrect.</td> </tr> </tbody> </table>	Value	Name	Description	0		AVC decoder will NOT go into MB concealment when Intra8x8/4x4 Prediction mode is incorrect.	1		AVC decoder will go into MB concealment when Intra8x8/4x4 Prediction mode is incorrect.
	Value	Name	Description								
	0		AVC decoder will NOT go into MB concealment when Intra8x8/4x4 Prediction mode is incorrect.								
	1		AVC decoder will go into MB concealment when Intra8x8/4x4 Prediction mode is incorrect.								
	0	<p>Intra Prediction Error Control Bit (applied to Intra16x16/Intra8x8/Intra4x4 Luma and Chroma)</p> <p>This field controls if AVC decoder will fix Intra Prediction Mode if the decoded value is incorrect according to MB position.</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>AVC decoder will detect and fix Intra Prediction Mode Errors.</td> </tr> <tr> <td>1</td> <td></td> <td>AVC decoder will retain the Intra Prediction value decoded from bitstream.</td> </tr> </tbody> </table>	Value	Name	Description	0		AVC decoder will detect and fix Intra Prediction Mode Errors.	1		AVC decoder will retain the Intra Prediction value decoded from bitstream.
Value	Name	Description									
0		AVC decoder will detect and fix Intra Prediction Mode Errors.									
1		AVC decoder will retain the Intra Prediction value decoded from bitstream.									

Inline Data Description - VP8 PAK OBJECT

Inline Data Description - VP8 PAK OBJECT														
Source:	VideoCS													
Size (in bits):	384													
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000													
This structure corresponds to Dw3..6 of MFX_VP8_PAK_OBJECT Command.														
DWord	Bit	Description												
0	31:23	Reserved Format: _____ MBZ												
	22:20	MV Format(Motion Vector Size) Exists If: _____ //IntraMbFlag = 0 This field specifies the size and format of the output motion vectors. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Intra MB</td> <td>No Motion vectors</td> </tr> <tr> <td>100b</td> <td>Inter Predict MB (Unpacked Motion Vector Mode)</td> <td>Sixteen Motion Vectors Per MacroBlock</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	000b	Intra MB	No Motion vectors	100b	Inter Predict MB (Unpacked Motion Vector Mode)	Sixteen Motion Vectors Per MacroBlock	Others	Reserved	
	Value	Name	Description											
	000b	Intra MB	No Motion vectors											
	100b	Inter Predict MB (Unpacked Motion Vector Mode)	Sixteen Motion Vectors Per MacroBlock											
	Others	Reserved												
			Programming Notes											
			This field MBZ, when the IntraMbFlag = 1 .											
	19:18	SegmentID Format: _____ U2 Segment number 0-3												
	17	Enable Coeff Clamp <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> <td>Magnitude of coefficients of the current MB is clamped based on the clamping matrix after quantization</td> </tr> <tr> <td>0</td> <td></td> <td>No Clamping</td> </tr> </tbody> </table>	Value	Name	Description	1		Magnitude of coefficients of the current MB is clamped based on the clamping matrix after quantization	0		No Clamping			
Value	Name	Description												
1		Magnitude of coefficients of the current MB is clamped based on the clamping matrix after quantization												
0		No Clamping												
16:14	Reserved Format: _____ MBZ													

Inline Data Description - VP8 PAK OBJECT																										
13	<p>Intra MB Flag This field specifies whether the current macroblock is an Intra (I) Macroblock. For Key pictures (IsKyeFrameFlag DW2, bit[5] of MFX_VP8_PIC_STATE), this field must be set to 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>0h</td> <td>INTER (Inter MacroBlock)</td> </tr> <tr> <td>1h</td> <td>INTRA (Intra MacroBlock)</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>For I-picture MB (Intra MB Flag = 1), this field must be set to 1.</p>		Value	Name	0h	INTER (Inter MacroBlock)	1h	INTRA (Intra MacroBlock)																		
Value	Name																									
0h	INTER (Inter MacroBlock)																									
1h	INTRA (Intra MacroBlock)																									
12:11	<p>RefPicSelect This field specifies which reference pic (among Last Frame, Golden Frame and Alt Frame) is selected for the current macroblock when Intra MB Flag = 0 .</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>00b</td> <td>Last Frame</td> </tr> <tr> <td>01b</td> <td>Golden Frame</td> </tr> <tr> <td>10b</td> <td>Alt Frame</td> </tr> </tbody> </table>		Value	Name	00b	Last Frame	01b	Golden Frame	10b	Alt Frame																
Value	Name																									
00b	Last Frame																									
01b	Golden Frame																									
10b	Alt Frame																									
10:8	<p>MB Type 3-Bits - Inter/Intra MB MB Type 3 Bits [10:8] specifies InterMB MV mode configurations: 16x16 or 2 16x8 or 4 8x8 or 16 4x4 when Intra MB Flag = 0 and bit [8] = IntraMB mode configurations: 4x4 or 16x16 when Intra MB Flag = 1</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>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>16x16</td> <td>Inter MB Only DW 6 bits 3:0 are used to indicate MVMode, MVMode can't be split</td> </tr> <tr> <td>001b</td> <td>2 16x8 (mv_Top Bottom)</td> <td>Inter MB [10:8] Split MV is inferred. DW5 bits[3:0] are used for MVMode for first 16x8 partition, DW6 bits[3:0] are used for MVMode for second 16x8 partition.</td> </tr> <tr> <td>010b</td> <td>2 8 x16 (mv_left_right)</td> <td>Inter MB [10:8] Split MV is inferred. DW5 bits[3:0] are used for MVMode for first 8x16 partition, DW5 bits[11:8] are used for MVMode for second 8x16 partition.</td> </tr> <tr> <td>011b</td> <td>4 8x8 (mv_quarters)</td> <td>Inter MB [10:8] Split MV is inferred. DW5 bits[3:0] are used for MVMode for first 8x8 partition. DW5 bits[11:8] are used for MvMode for second 8x8 partition. DW6 bits[3:0] are used for MVMode for third 8x8 partition. DW6 bits[11:8] are used for MVMode for fourth 8x8 partition.</td> </tr> <tr> <td>100b</td> <td>16 4x4 (mv_16)</td> <td>Inter MB [10:8] Split MV is inferred. There are 16 partitions. Each Sub-block uses 4 bits in DW6 and DW7.</td> </tr> <tr> <td>0b</td> <td>16x16</td> <td>Intra MB [8] Only DW5, bits[3:0] are used for Y mode. For B_PRED, "16 4x4" should be used which implies B_PRED mode.</td> </tr> <tr> <td>1b</td> <td>16 4x4</td> <td>Intra MB [8] All bits in DW5 and DW6 are used to represent B_PRED modes (Bmodes) in each sub-blocks.</td> </tr> </tbody> </table>		Value	Name	Description	000b	16x16	Inter MB Only DW 6 bits 3:0 are used to indicate MVMode, MVMode can't be split	001b	2 16x8 (mv_Top Bottom)	Inter MB [10:8] Split MV is inferred. DW5 bits[3:0] are used for MVMode for first 16x8 partition, DW6 bits[3:0] are used for MVMode for second 16x8 partition.	010b	2 8 x16 (mv_left_right)	Inter MB [10:8] Split MV is inferred. DW5 bits[3:0] are used for MVMode for first 8x16 partition, DW5 bits[11:8] are used for MVMode for second 8x16 partition.	011b	4 8x8 (mv_quarters)	Inter MB [10:8] Split MV is inferred. DW5 bits[3:0] are used for MVMode for first 8x8 partition. DW5 bits[11:8] are used for MvMode for second 8x8 partition. DW6 bits[3:0] are used for MVMode for third 8x8 partition. DW6 bits[11:8] are used for MVMode for fourth 8x8 partition.	100b	16 4x4 (mv_16)	Inter MB [10:8] Split MV is inferred. There are 16 partitions. Each Sub-block uses 4 bits in DW6 and DW7.	0b	16x16	Intra MB [8] Only DW5, bits[3:0] are used for Y mode. For B_PRED, "16 4x4" should be used which implies B_PRED mode.	1b	16 4x4	Intra MB [8] All bits in DW5 and DW6 are used to represent B_PRED modes (Bmodes) in each sub-blocks.
Value	Name	Description																								
000b	16x16	Inter MB Only DW 6 bits 3:0 are used to indicate MVMode, MVMode can't be split																								
001b	2 16x8 (mv_Top Bottom)	Inter MB [10:8] Split MV is inferred. DW5 bits[3:0] are used for MVMode for first 16x8 partition, DW6 bits[3:0] are used for MVMode for second 16x8 partition.																								
010b	2 8 x16 (mv_left_right)	Inter MB [10:8] Split MV is inferred. DW5 bits[3:0] are used for MVMode for first 8x16 partition, DW5 bits[11:8] are used for MVMode for second 8x16 partition.																								
011b	4 8x8 (mv_quarters)	Inter MB [10:8] Split MV is inferred. DW5 bits[3:0] are used for MVMode for first 8x8 partition. DW5 bits[11:8] are used for MvMode for second 8x8 partition. DW6 bits[3:0] are used for MVMode for third 8x8 partition. DW6 bits[11:8] are used for MVMode for fourth 8x8 partition.																								
100b	16 4x4 (mv_16)	Inter MB [10:8] Split MV is inferred. There are 16 partitions. Each Sub-block uses 4 bits in DW6 and DW7.																								
0b	16x16	Intra MB [8] Only DW5, bits[3:0] are used for Y mode. For B_PRED, "16 4x4" should be used which implies B_PRED mode.																								
1b	16 4x4	Intra MB [8] All bits in DW5 and DW6 are used to represent B_PRED modes (Bmodes) in each sub-blocks.																								

Inline Data Description - VP8 PAK OBJECT											
	7:6 Reserved Format: _____ MBZ										
	5:4 MB UV 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;">0</td> <td>DC_PRED</td> </tr> <tr> <td style="text-align: center;">1</td> <td>V_PRED</td> </tr> <tr> <td style="text-align: center;">2</td> <td>H_PRED</td> </tr> <tr> <td style="text-align: center;">3</td> <td>TM_PRED</td> </tr> </tbody> </table>	Value	Name	0	DC_PRED	1	V_PRED	2	H_PRED	3	TM_PRED
	Value	Name									
	0	DC_PRED									
	1	V_PRED									
2	H_PRED										
3	TM_PRED										
3 Reserved Format: _____ MBZ											
2 Skip MB Flag This field is equivalent to mb_skip_flag in VP8 spec. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>By setting this field to 1, it forces an Inter MacroBlock to be encoded as a skipped MacroBlock</td> </tr> </tbody> </table>	Programming Notes	By setting this field to 1, it forces an Inter MacroBlock to be encoded as a skipped MacroBlock									
Programming Notes											
By setting this field to 1, it forces an Inter MacroBlock to be encoded as a skipped MacroBlock											
1:0 Reserved Format: _____ MBZ											
1	31:24 Reserved Format: _____ MBZ										
	23:16 MbYCnt (Vertical Origin) Format: _____ U8 Unit of MacroBlock This field specifies the vertical origin of current macroblock in the destination picture in units of macroblocks.										
	15:8 Reserved Format: _____ MBZ										
	7:0 MbXCnt (Horizontal Origin) Format: _____ U8 Unit of MacroBlock This field specifies the horizontal origin of current macroblock in the destination picture in units of macroblocks.										
2	31:28 B Mode for SubBlock7 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.										
	27:24 B Mode for SubBlock6 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.										
	23:20 B Mode for SubBlock5 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.										
	19:16 B Mode for SubBlock4 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.										

Inline Data Description - VP8 PAK OBJECT			
	15:12 B Mode for SubBlock3 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	11:8 B Mode for SubBlock2 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	7:4 B Mode for SubBlock1 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	3:0 B Mode for SubBlock0 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
3	31:28 B Mode for SubBlock15 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	27:24 B Mode for SubBlock14(Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	23:20 B Mode for SubBlock13(Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	19:16 B Mode for SubBlock12(Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	15:12 B Mode for SubBlock11(Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	11:8 B Mode for SubBlock10 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	7:4 B Mode for SubBlock9 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
	3:0 B Mode for SubBlock8 (Y mode for the macroblock in non-B mode) For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.		
4	31:16 MV Y FWD 0 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U16</td></tr></table> The value of the y component of this motion vector for FWD block 0.		U16
		U16	
15:0 MV X FWD 0 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U16</td></tr></table> The value of the x component of this motion vector for FWD block 0.		U16	
	U16		
5	31:16 MV Y FWD 1 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U16</td></tr></table> The value of the y component of this motion vector for FWD block 1.		U16
		U16	
15:0 MV X FWD 1 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U16</td></tr></table> The value of the x component of this motion vector for FWD block 1.		U16	
	U16		
6	31:16 MV Y FWD 2 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td style="width: 50px;">U16</td></tr></table> The value of the y component of this motion vector for FWD block 2.		U16
	U16		

Inline Data Description - VP8 PAK OBJECT		
	15:0	MV X FWD 2
		Format: U16 The value of the x component of this motion vector for FWD block 2.
7	31:16	MV Y FWD 3
		Format: U16 The value of the y component of this motion vector for FWD block 3.
	15:0	MV X FWD 3
		Format: U16 The value of the x component of this motion vector for FWD block 3.
8	31:16	MV Y BWD 0
		Format: U16 The value of the y component of this motion vector for BWD block 0.
	15:0	MV X BWD 0
		Format: U16 The value of the x component of this motion vector for BWD block 0.
9	31:16	MV Y BWD 1
		Format: U16 The value of the y component of this motion vector for BWD block 1.
	15:0	MV X BWD 1
		Format: U16 The value of the x component of this motion vector for BWD block 1.
10	31:16	MV Y BWD 2
		Format: U16 The value of the y component of this motion vector for BWD block 2.
	15:0	MV X BWD 2
		Format: U16 The value of the x component of this motion vector for BWD block 2.
11	31:16	MV Y BWD 3
		Format: U16 The value of the y component of this motion vector for BWD block 3.
	15:0	MV X BWD 3
		Format: U16 The value of the x component of this motion vector for BWD block 3.

INTERFACE_DESCRIPTOR_DATA

INTERFACE_DESCRIPTOR_DATA											
Source:	RenderCS										
Size (in bits):	256										
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000										
DWord	Bit	Description									
0	31:6	<p>Kernel Start Pointer</p> <table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[31:6]Kernel</td> </tr> </table> <p>Specifies the 64-byte aligned address offset of the first instruction in the kernel. This pointer is relative to the Instruction Base Address.</p>	Format:	InstructionBaseOffset[31:6]Kernel							
	Format:	InstructionBaseOffset[31:6]Kernel									
5:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
1	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
15:0	<p>Kernel Start Pointer High</p> <table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[47:32]Kernel</td> </tr> </table> <p>This field specifies the high 16 bits of starting address of the Kernel Pointer.</p>	Format:	InstructionBaseOffset[47:32]Kernel								
Format:	InstructionBaseOffset[47:32]Kernel										
2	31:20	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
	Format:	MBZ									
	19	<p>Denorm Mode</p> <p>This field specifies how Float denormalized numbers are handles in the dispatched thread.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Ftz</td> <td>Float denorms will be flushed to zero when appearing as inputs, denorms will never come out of instructions. Double precision float and half precision float numbers are not flushed to zero.</td> </tr> <tr> <td>1h</td> <td>SetByKernel</td> <td>Denorms will be handled in by kernel.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Ftz	Float denorms will be flushed to zero when appearing as inputs, denorms will never come out of instructions. Double precision float and half precision float numbers are not flushed to zero.	1h	SetByKernel	Denorms will be handled in by kernel.
	Value	Name	Description								
0h	Ftz	Float denorms will be flushed to zero when appearing as inputs, denorms will never come out of instructions. Double precision float and half precision float numbers are not flushed to zero.									
1h	SetByKernel	Denorms will be handled in by kernel.									
18	<p>Single Program Flow</p> <p>Specifies whether the kernel program has a single program flow (SIMDn_{xm} with m = 1) or multiple program flows (SIMDn_{xm} with m > 1).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Multiple</td> </tr> <tr> <td>1h</td> <td>Single</td> </tr> </tbody> </table>	Value	Name	0h	Multiple	1h	Single				
Value	Name										
0h	Multiple										
1h	Single										
17	<p>Thread Priority</p> <p>Specifies the priority of the thread for dispatch.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Normal Priority</td> </tr> <tr> <td>1h</td> <td>High Priority</td> </tr> </tbody> </table>	Value	Name	0h	Normal Priority	1h	High Priority				
Value	Name										
0h	Normal Priority										
1h	High Priority										

INTERFACE_DESCRIPTOR_DATA												
16	<p>Floating Point Mode Specifies the floating point mode used by the dispatched thread.</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>IEEE-754</td> </tr> <tr> <td>1h</td> <td>Alternate</td> </tr> </tbody> </table>	Value	Name	0h	IEEE-754	1h	Alternate					
	Value	Name										
	0h	IEEE-754										
	1h	Alternate										
	15:14	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
	13	<p>Illegal Opcode Exception Enable</p> <table border="1"> <tr> <td>Format:</td> <td>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>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
	Format:	MBZ										
11	<p>Mask Stack Exception Enable</p> <table border="1"> <tr> <td>Format:</td> <td>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"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
7	<p>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 <i>Exceptions and ISA Execution Environment</i>.</p>	Format:	Enable									
Format:	Enable											
6:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
3	<p>31:5 Sampler State Pointer</p> <table border="1"> <tr> <td>Format:</td> <td>DynamicStateOffset[31:5]SAMPLER_STATE</td> </tr> </table> <p>Specifies the 32-byte aligned address offset of the sampler state table. This pointer is relative to the Dynamic State Base Address. <i>This field is ignored for child threads.</i></p>	Format:	DynamicStateOffset[31:5]SAMPLER_STATE									
	Format:	DynamicStateOffset[31:5]SAMPLER_STATE										
<p>4:2 Sampler Count</p> <table border="1"> <tr> <td>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. <i>This field is ignored for child threads. If this field is not zero, sampler state is prefetched for the first instance of a root thread upon the startup of the media pipeline.</i></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>[0,4]</td> <td></td> </tr> <tr> <td>0h</td> <td>No samplers used</td> </tr> <tr> <td>1h</td> <td>Between 1 and 4 samplers used</td> </tr> <tr> <td>2h</td> <td>Between 5 and 8 samplers used</td> </tr> </tbody> </table>	Format:	U3	Value	Name	[0,4]		0h	No samplers used	1h	Between 1 and 4 samplers used	2h	Between 5 and 8 samplers used
Format:	U3											
Value	Name											
[0,4]												
0h	No samplers used											
1h	Between 1 and 4 samplers used											
2h	Between 5 and 8 samplers used											

INTERFACE_DESCRIPTOR_DATA										
		<table border="1"> <tr> <td>3h</td> <td>Between 9 and 12 samplers used</td> </tr> <tr> <td>4h</td> <td>Between 13 and 16 samplers used</td> </tr> </table>	3h	Between 9 and 12 samplers used	4h	Between 13 and 16 samplers used				
3h	Between 9 and 12 samplers used									
4h	Between 13 and 16 samplers used									
	1:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ									
4	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
	Format:	MBZ								
	15:5	<p>Binding Table Pointer</p> <table border="1"> <tr> <td>Format:</td> <td>SurfaceStateOffset[15:5]BINDING_TABLE_STATE*256</td> </tr> </table> <p style="text-align: center;">Description</p> <p>Specifies the 32-byte aligned address of the binding table. This pointer is relative to the Surface State Base Address. <i>This field is ignored for child threads.</i></p>	Format:	SurfaceStateOffset[15:5]BINDING_TABLE_STATE*256						
	Format:	SurfaceStateOffset[15:5]BINDING_TABLE_STATE*256								
4:0	<p>Binding Table Entry Count</p> <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Specifies how many binding table entries the kernel uses. Used only for prefetching of the binding table entries and associated surface state. <i>This field is ignored for child threads. If this field is not zero, binding table and surface state are prefetched for the first instance of a root thread upon the startup of the media pipeline.</i></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>[0,31]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>The maximum number of prefetched binding table entries is limited to 31. 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.</p>	Format:	U5	Value	Name	[0,31]				
Format:	U5									
Value	Name									
[0,31]										
5	31:16	<p>Constant/Indirect URB Entry Read Length</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>Specifies the amount of URB data read and passed in the thread payload for the Constant or Indirect URB entry, in 8-DW register increments. A value 0 means that no Constant or Indirect URB Entry will be loaded. The Constant URB Entry Read Offset field will then be ignored. In GPGPU mode this describes how much data is delivered in a single dispatch. Multiple dispatches in a thread group will deliver constant data offset by this value. The total amount of constant data is (Constant URB Read Length * Number of Threads in GPGPU Thread Group + Cross-Thread Constant Data Read Length).</p> <p>If Cross-Thread Constant Data Read Length for Indirect is greater than 0, then this field must also be greater than 0. The allowed combinations are:</p> <table border="1"> <thead> <tr> <th>Constant/Indirect URB Entry Read Length</th> <th>Cross-Thread Constant Data Read Length</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>=0</td> <td>=0</td> <td>No Payload</td> </tr> </tbody> </table>	Format:	U16	Constant/Indirect URB Entry Read Length	Cross-Thread Constant Data Read Length	Notes	=0	=0	No Payload
Format:	U16									
Constant/Indirect URB Entry Read Length	Cross-Thread Constant Data Read Length	Notes								
=0	=0	No Payload								

INTERFACE_DESCRIPTOR_DATA																
		<table border="1"> <tr> <td>>0</td> <td>=0</td> <td>Per-thread payload only</td> </tr> <tr> <td>>0</td> <td>>0</td> <td>Both kinds of payload</td> </tr> <tr> <td>=0</td> <td>>0</td> <td>Only for CURBE payloads</td> </tr> </table>	>0	=0	Per-thread payload only	>0	>0	Both kinds of payload	=0	>0	Only for CURBE payloads					
>0	=0	Per-thread payload only														
>0	>0	Both kinds of payload														
=0	>0	Only for CURBE payloads														
		<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,63]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,63]											
Value	Name															
[0,63]																
15:0	Constant URB Entry Read Offset															
	Format:	U16														
	Specifies the offset (in 8-DW units) at which Constant URB data is to be read from the URB before being included in the thread payload.															
	<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>[0,1983]</td> <td></td> <td>Indicating [0,1983] 256-bit register increments. ROB has 64KB of storage; 2048 entries. However, lowest 64 entries are reserved for VFE/TS to store interface descriptor data. Hence, (URB Entry Read Offset + Read Length) shall not exceed 1984.</td> </tr> </tbody> </table>	Value	Name	Description	[0,1983]		Indicating [0,1983] 256-bit register increments. ROB has 64KB of storage; 2048 entries. However, lowest 64 entries are reserved for VFE/TS to store interface descriptor data. Hence, (URB Entry Read Offset + Read Length) shall not exceed 1984.									
Value	Name	Description														
[0,1983]		Indicating [0,1983] 256-bit register increments. ROB has 64KB of storage; 2048 entries. However, lowest 64 entries are reserved for VFE/TS to store interface descriptor data. Hence, (URB Entry Read Offset + Read Length) shall not exceed 1984.														
6	31:24	Reserved														
	Format:	MBZ														
	23:22	Rounding Mode														
	Format:	U2														
	<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>00b</td> <td>RTNE [Default]</td> <td>Round to Nearest Even</td> </tr> <tr> <td>01b</td> <td>RU</td> <td>Round toward +Infinity</td> </tr> <tr> <td>10b</td> <td>RD</td> <td>Round toward -Infinity</td> </tr> <tr> <td>11b</td> <td>RTZ</td> <td>Round toward Zero</td> </tr> </tbody> </table>	Value	Name	Description	00b	RTNE [Default]	Round to Nearest Even	01b	RU	Round toward +Infinity	10b	RD	Round toward -Infinity	11b	RTZ	Round toward Zero
Value	Name	Description														
00b	RTNE [Default]	Round to Nearest Even														
01b	RU	Round toward +Infinity														
10b	RD	Round toward -Infinity														
11b	RTZ	Round toward Zero														
	21	Barrier Enable														
	Format:	Enable														
	This field specifies whether the thread group requires a barrier. If not, it can be dispatched without allocating one.															
	20:16	Shared Local Memory Size														
	Format:	U5														
	This field indicates how much Shared Local Memory the thread group requires. The amount is specified in 4k blocks, but only powers of 2 are allowed: 0, 4k, 8k, 16k, 32k and 64k per half-slice.															

INTERFACE_DESCRIPTOR_DATA																													
		Uses a different encoding to allow encodings for the new 1k and 2k SLM sizes.																											
		<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</td> <td>Encodes 0K</td> <td>No SLM used</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Encodes 1K</td> <td></td> </tr> <tr> <td style="text-align: center;">2</td> <td>Encodes 2K</td> <td></td> </tr> <tr> <td style="text-align: center;">3</td> <td>Encodes 4K</td> <td></td> </tr> <tr> <td style="text-align: center;">4</td> <td>Encodes 8K</td> <td></td> </tr> <tr> <td style="text-align: center;">5</td> <td>Encodes 16K</td> <td></td> </tr> <tr> <td style="text-align: center;">6</td> <td>Encodes 32K</td> <td></td> </tr> <tr> <td style="text-align: center;">7</td> <td>Encodes 64K</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0	Encodes 0K	No SLM used	1	Encodes 1K		2	Encodes 2K		3	Encodes 4K		4	Encodes 8K		5	Encodes 16K		6	Encodes 32K		7	Encodes 64K	
Value	Name	Description																											
0	Encodes 0K	No SLM used																											
1	Encodes 1K																												
2	Encodes 2K																												
3	Encodes 4K																												
4	Encodes 8K																												
5	Encodes 16K																												
6	Encodes 32K																												
7	Encodes 64K																												
	15	<p>Global Barrier Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p style="text-align: center;">Description</p> <p>This field when set indicates that the thread group associated with this barrier is allowed to cross sub-slices with a performance penalty. When this field is clear, the thread group is dispatched to a single sub-slice or pool. Note that SLM should not be used with a Global Barrier since SLM is always forced to a single sub-slice or pool. The Barrier Enable bit must be set to enable the barrier, since this bit only specifies the type of barrier.</p> <p>Restriction: Global barriers cannot be used. Must be zero.</p>	Format:	Enable																									
Format:	Enable																												
	14:10	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																									
Format:	MBZ																												
	9:0	<p>Number of Threads in GPGPU Thread Group</p> <table border="1"> <tr> <td>Format:</td> <td>U10</td> </tr> </table> <p>Specifies the number of threads that are in this thread group. The minimum value is 1, while the maximum value is the number of threads in a subslice for local barriers. See vol1b Configurations for the number of threads per subslice for different products. The maximum value for global barriers is limited by the number of threads in the system, or by 511, whichever is lower. This field should not be set to 0 even if the barrier is disabled, since an accurate value is needed for proper pre-emption.</p>	Format:	U10																									
Format:	U10																												
7	31:8	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																									
Format:	MBZ																												

INTERFACE_DESCRIPTOR_DATA							
7:0	<p>Cross-Thread Constant Data Read Length</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Specifies the amount of constant data in CURBE in 8-DW register increments which will be sent to every thread in the thread group in addition to the per thread ids specified by Constant URB Entry Read Length.</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>[0,127]</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	[0,127]	
Format:	U8						
Value	Name						
[0,127]							

INTERRUPT

INTERRUPT	
Source:	BSpec
Access:	RO, R/W, R/WC, R/W
Size (in bits):	128
Default Value:	0x00000000, 0xFFFFFFFF, 0x00000000, 0x00000000

See the Interrupt Definition Tables to find the source event for each interrupt bit. There are multiple instances of this register format.

DWord	Bit	Description										
0	31:0	<p>ISR</p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> </table> <p>These are the Interrupt Status Register Bits. This field contains the non-persistent values of the interrupt status bits. The IMR selects which of these interrupt conditions are reported in the persistent IIR</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Condition Doesn't exist</td> </tr> <tr> <td>1b</td> <td>Condition Exists</td> </tr> </tbody> </table> <p style="text-align: center;">Restriction</p> <p>Some inputs to this register are short pulses. Do not use this register to sample these conditions.</p>	Access:	RO	Value	Name	0b	Condition Doesn't exist	1b	Condition Exists		
Access:	RO											
Value	Name											
0b	Condition Doesn't exist											
1b	Condition Exists											
1	31:0	<p>IMR</p> <table border="1"> <tr> <td>Access:</td> <td>R/W</td> </tr> </table> <p>These are the Interrupt Mask Register Bits. This field contains a bit mask which selects which interrupt bits from the ISR are reported in the IIR.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>FFFFFFFFh</td> <td>All interrupts masked [Default]</td> </tr> <tr> <td>0b</td> <td>Not Masked</td> </tr> <tr> <td>1b</td> <td>Masked</td> </tr> </tbody> </table> <p style="text-align: center;">Restriction</p> <p>For GT interrupts DO NOT use this register to mask interrupt events. Instead program this IMR to all 0s and use the individual GT command streamer MASK bits in the GT register space. This prevents unneeded messaging to DE.</p>	Access:	R/W	Value	Name	FFFFFFFFh	All interrupts masked [Default]	0b	Not Masked	1b	Masked
Access:	R/W											
Value	Name											
FFFFFFFFh	All interrupts masked [Default]											
0b	Not Masked											
1b	Masked											
2	31:0	<p>IIR</p> <table border="1"> <tr> <td>Access:</td> <td>R/WC</td> </tr> </table> <p>These are the Interrupt Identity Register Bits. This field holds the persistent values of the interrupt bits from the ISR which are unmasked by the IMR. The IER enables an interrupt to be generated when the corresponding bit in the IIR becomes set. A disabled interrupt will still appear in the IIR. Bits set in this register will remain set (persist) until the interrupt condition is cleared by writing a '1' to the appropriate bits.</p>	Access:	R/WC								
Access:	R/WC											

INTERRUPT													
	<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>Condition Not Detected</td> </tr> <tr> <td>1b</td> <td>Condition Detected</td> </tr> </tbody> </table>	Value	Name	0b	Condition Not Detected	1b	Condition Detected						
Value	Name												
0b	Condition Not Detected												
1b	Condition Detected												
	<p style="text-align: center;">Programming Notes</p> <p>For each bit, the IIR can store a second pending interrupt if two or more of the same interrupt conditions occur before the first condition is cleared. Upon clearing the first interrupt, the IIR bit will momentarily go low, then return high to indicate there is second interrupt pending.</p>												
3	<table border="1"> <tr> <td>31:0</td> <td>IER</td> </tr> <tr> <td>Access:</td> <td>R/W</td> </tr> <tr> <td colspan="2">These are the Interrupt Enable Register Bits. The field enables an interrupt to be generated when the corresponding bit in the IIR becomes set. A disabled interrupt will still appear in the IIR.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>0b</td> <td>Disabled</td> </tr> <tr> <td>1b</td> <td>Enabled</td> </tr> </table>	31:0	IER	Access:	R/W	These are the Interrupt Enable Register Bits. The field enables an interrupt to be generated when the corresponding bit in the IIR becomes set. A disabled interrupt will still appear in the IIR.		Value	Name	0b	Disabled	1b	Enabled
31:0	IER												
Access:	R/W												
These are the Interrupt Enable Register Bits. The field enables an interrupt to be generated when the corresponding bit in the IIR becomes set. A disabled interrupt will still appear in the IIR.													
Value	Name												
0b	Disabled												
1b	Enabled												
	<p style="text-align: center;">Programming Notes</p> <p>The master interrupt enable must be set to 1b for any of these enabled interrupts to propagate to PCI device 2 interrupt processing.</p>												

Invalidate After Read Message Descriptor Control Field

MDC_IAR - Invalidate After Read Message Descriptor Control Field				
Source:	BSpec			
Size (in bits):	1			
Default Value:	0x00000000			
DWord	Bit	Description		
0	0	<p>Reserved</p> <table border="1" data-bbox="305 611 1471 657"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Previously, this Enable field was intended to optimize scratch and spill/fill read messages, where the memory was only used by a single thread and did not need to be maintained after the thread completed. If enabled, it caused all lines in the L3 cache accessed by the message to be invalidated after the read occurred, regardless of whether the line contained modified data. It was intended as a performance hint indicating that the data would no longer be used to avoid writing back data to memory.</p>	Format:	MBZ
Format:	MBZ			

JPEG

JPEG				
Source:	VideoCS			
Size (in bits):	16			
Default Value:	0x00000000			
DWord	Bit	Description		
0	15:5	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		
	4	Inconsistent VLD SE Error This flag indicates an inconsistent SE coded in the bit-stream. Bit-stream does not match any entries in the hauffman table.		
	3	Extra Block Error This flag indicates extra block coded within an ECS data boundary.		
	2	Missing block Error This flag indicates one or more blocks are missing within an ECS data boundary.		
	1	Extra ECS Error This flag indicates extra ECS' coded in the bit-stream SCAN payload data.		
0	Missing ECS Error This flag indicates one or more ECS' are missing from the bit-stream SCAN payload data.			

LOD Message Address Payload Control

MACD_LOD - LOD Message Address Payload Control							
Source:	BSpec						
Size (in bits):	32						
Default Value:	0x00000000						
DWord	Bit	Description					
0	31:4	Reserved Format: MBZ Ignored					
	3:0	LOD Format: U4 Specifies the LOD for this slot. <table border="1" data-bbox="535 856 1469 945"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,14]</td> <td></td> <td>representing LOD</td> </tr> </tbody> </table>	Value	Name	Description	[0,14]	
Value	Name	Description					
[0,14]		representing LOD					

Lower Oword Block Data Payload

MDP_OW1L - Lower Oword Block Data Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.3	127:0	Oword
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U128</td> </tr> </table> <p>Specifies the upper Oword data element</p>
Format:	U128	
0.4-0.7	127:0	Reserved
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>
Format:	Ignore	

MEDIA_SURFACE_STATE

DWord		Bit	Description										
Source:			BSpec										
Exists If:			//([MessageType] == 'Deinterlace') OR ([MessageType] == 'Sample_8x8')										
Size (in bits):			256										
Default Value:			0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000										
This is the SURFACE_STATE used by only deinterlace, sample_8x8, and VME messages.													
0	31:30	Rotation <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No Rotation or 0 Degree</td> </tr> <tr> <td>01b</td> <td>90 Degree Rotation</td> </tr> <tr> <td>10b</td> <td>180 Degree Rotation</td> </tr> <tr> <td>11b</td> <td>270 Degree Rotation</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> Rotation is only supported only with AVS function messages and not with HDC direct write and 16x8 AVS messages.		Value	Name	00b	No Rotation or 0 Degree	01b	90 Degree Rotation	10b	180 Degree Rotation	11b	270 Degree Rotation
Value	Name												
00b	No Rotation or 0 Degree												
01b	90 Degree Rotation												
10b	180 Degree Rotation												
11b	270 Degree Rotation												
	29:27	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ								
Format:	MBZ												
	26:20	X Offset <table border="1"> <tr> <td>Exists If:</td> <td>//[Surface Format] is one of Planar Formats</td> </tr> <tr> <td>Format:</td> <td>PixelFormat[8:2]</td> </tr> </table> <p>This field specifies the horizontal offset in pixels from the Surface Base Address to the start (origin) of the surface. This field effectively loosens the alignment restrictions on the origin of tiled surfaces. Previously, tiled surface origin was (by definition) located at the base address, and thus needed to satisfy the 4KB base address alignment restriction. Now the origin can be specified at a finer (4-wide x 4-high pixel) resolution.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,508]</td> <td></td> <td>In multiples of 4 (low 2 bits missing)</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> For linear surfaces and Packed Formats, this field must be zero. For Surface Format with 8 bits per element, this field must be a multiple of 16. For Surface Format with 16 bits per element, this field must be a multiple of 8.		Exists If:	//[Surface Format] is one of Planar Formats	Format:	PixelFormat[8:2]	Value	Name	Description	[0,508]		In multiples of 4 (low 2 bits missing)
Exists If:	//[Surface Format] is one of Planar Formats												
Format:	PixelFormat[8:2]												
Value	Name	Description											
[0,508]		In multiples of 4 (low 2 bits missing)											

MEDIA_SURFACE_STATE										
	26:16	Reserved								
		Exists If: // [Surface Format] is not one of Planar Formats								
		Format: MBZ								
	19:16	Y Offset								
		Exists If: // [Surface Format] is one of Planar Formats								
	Format: RowOffset[5:2]									
	This field specifies the vertical offset in rows from the Surface Base Address to the start of the surface. (See additional description in the X Offset field)									
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,28]</td> <td></td> <td>In multiples of 4 (low two bits missing)</td> </tr> </tbody> </table>		Value	Name	Description	[0,28]		In multiples of 4 (low two bits missing)		
Value	Name	Description								
[0,28]		In multiples of 4 (low two bits missing)								
	Programming Notes									
	For linear surfaces and Packed Formats, this field must be zero.									
	15:12	Reserved								
		Format: MBZ								
	11:0	Reserved								
		Format: MBZ								
1	31:18	Height								
		Format: U14-1								
	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.									
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> <th>Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> <td>representing heights [1,16384]</td> <td>[Surface Type] != FM_STRBUF_*</td> </tr> </tbody> </table>		Value	Name	Description	Exists If	[0,16383]		representing heights [1,16384]	[Surface Type] != FM_STRBUF_*
Value	Name	Description	Exists If							
[0,16383]		representing heights [1,16384]	[Surface Type] != FM_STRBUF_*							
	Programming Notes									
	Height (field value + 1) must be a multiple of 2 for PLANAR_420 surfaces. If Vertical Line Stride is 1, this field indicates the height of the field, not the height of the frame.									

MEDIA_SURFACE_STATE											
17:4	<p>Width</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 width of the surface in units of pixels. For PLANAR surface formats, this field indicates the width of the Y (luma) plane.</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: 30%;">Description</th> <th style="width: 30%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> <td>representing widths [1,16384]</td> <td>[Surface Type] != FM_STRBUF_*</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <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, PLANAR_422, and all YCRCB_* and Y16_UNORM surfaces, and must be a multiple of 4 for PLANAR_411 and Y8_UNORM_VA surfaces. For deinterlace messages, the Width (field value + 1) must be a multiple of 8. <ul style="list-style-type: none"> For Y8_UNORM_VA format width should be in multiple of 4, for Y16_UNORM_VA format width should be in multiple of 2, for Y1_UNORM format width should be in multiple of 32 When Address Control = Mirror, the total width should be in multiple of 4bytes. <p>Width (field value + 1) must be a multiple of 2 for PLANAR_420_16</p>	Format:	U14-1	Value	Name	Description	Exists If	[0,16383]		representing widths [1,16384]	[Surface Type] != FM_STRBUF_*
	Format:	U14-1									
Value	Name	Description	Exists If								
[0,16383]		representing widths [1,16384]	[Surface Type] != FM_STRBUF_*								
3:2	<p>Picture Structure</p> <p>Specifies the encoding of the current picture.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 70%;">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>10b</td> <td>Bottom Field Picture</td> </tr> <tr> <td>11b</td> <td>Invalid, not allowed</td> </tr> </tbody> </table>	Value	Name	00b	Frame Picture	01b	Top Field Picture	10b	Bottom Field Picture	11b	Invalid, not allowed
Value	Name										
00b	Frame Picture										
01b	Top Field Picture										
10b	Bottom Field Picture										
11b	Invalid, not allowed										

MEDIA_SURFACE_STATE																																																											
1:0	Cr(V)/Cb(U) Pixel Offset V Direction																																																										
	Default Value:	0																																																									
	Format:	U0.2																																																									
	Description																																																										
	Specifies the distance to the U/V values with respect to the even numbered Y channels in the V direction																																																										
	Programming Notes																																																										
This field is ignored for all formats except PLANAR_420_8																																																											
This offset has been increased from 2 bits to 3 bits to support U1.2 format, and the MSB bit is added as Pixel Offset V Direction MSB in DWord 2. Valid values for the combined field range from 0 to 4.																																																											
2	31:27	<p>Surface Format</p> <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> <p>Note: Y8_UNORM_VA, Y16_UNORM and Y16_SNORM are used for all functions of sample_8x8 except AVS where rest of the formats are not used. These two formats are packed as 32bits in L1 though the individual pixels are either 8bpp or 16bpp respectively.</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>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></td></tr> <tr><td>5</td><td>Y8_UNORM_VA</td><td>Sample_8x8 only except AVS</td></tr> <tr><td>6</td><td>Y16_SNORM</td><td>Sample_8x8 only except AVS</td></tr> <tr><td>7</td><td>Y16_UNORM_VA</td><td>Sample_8x8 only except AVS</td></tr> <tr><td>8</td><td>R10G10B10A2_UNORM</td><td>Sample_8x8 only</td></tr> <tr><td>9</td><td>R8G8B8A8_UNORM</td><td>Sample_8x8 AVS only</td></tr> <tr><td>10</td><td>R8B8_UNORM (CrCb)</td><td>Sample_8x8 AVS only</td></tr> <tr><td>11</td><td>R8_UNORM (Cr/Cb)</td><td>Sample_8x8 AVS only</td></tr> <tr><td>12</td><td>Y8_UNORM</td><td>Sample_8x8 AVS only</td></tr> <tr><td>13</td><td>A8Y8U8V8_UNORM</td><td>Sample_8x8 AVS only</td></tr> <tr><td>14</td><td>B8G8R8A8_UNORM</td><td>Sample_8x8 AVS only</td></tr> <tr><td>15</td><td>R16G16B16A16</td><td>Sample_8x8 AVS only</td></tr> <tr><td>16</td><td>Y1_UNORM</td><td>Sample_8x8 only for boolean surfaces (1bit/pixel)</td></tr> <tr><td>18</td><td>PLANAR_422_8</td><td>Sample_8x8 AVS only</td></tr> </tbody> </table>	Value	Name	Description	0	YCRCB_NORMAL		1	YCRCB_SWAPUVY		2	YCRCB_SWAPUV		3	YCRCB_SWAPY		4	PLANAR_420_8		5	Y8_UNORM_VA	Sample_8x8 only except AVS	6	Y16_SNORM	Sample_8x8 only except AVS	7	Y16_UNORM_VA	Sample_8x8 only except AVS	8	R10G10B10A2_UNORM	Sample_8x8 only	9	R8G8B8A8_UNORM	Sample_8x8 AVS only	10	R8B8_UNORM (CrCb)	Sample_8x8 AVS only	11	R8_UNORM (Cr/Cb)	Sample_8x8 AVS only	12	Y8_UNORM	Sample_8x8 AVS only	13	A8Y8U8V8_UNORM	Sample_8x8 AVS only	14	B8G8R8A8_UNORM	Sample_8x8 AVS only	15	R16G16B16A16	Sample_8x8 AVS only	16	Y1_UNORM	Sample_8x8 only for boolean surfaces (1bit/pixel)	18	PLANAR_422_8	Sample_8x8 AVS only
Value	Name	Description																																																									
0	YCRCB_NORMAL																																																										
1	YCRCB_SWAPUVY																																																										
2	YCRCB_SWAPUV																																																										
3	YCRCB_SWAPY																																																										
4	PLANAR_420_8																																																										
5	Y8_UNORM_VA	Sample_8x8 only except AVS																																																									
6	Y16_SNORM	Sample_8x8 only except AVS																																																									
7	Y16_UNORM_VA	Sample_8x8 only except AVS																																																									
8	R10G10B10A2_UNORM	Sample_8x8 only																																																									
9	R8G8B8A8_UNORM	Sample_8x8 AVS only																																																									
10	R8B8_UNORM (CrCb)	Sample_8x8 AVS only																																																									
11	R8_UNORM (Cr/Cb)	Sample_8x8 AVS only																																																									
12	Y8_UNORM	Sample_8x8 AVS only																																																									
13	A8Y8U8V8_UNORM	Sample_8x8 AVS only																																																									
14	B8G8R8A8_UNORM	Sample_8x8 AVS only																																																									
15	R16G16B16A16	Sample_8x8 AVS only																																																									
16	Y1_UNORM	Sample_8x8 only for boolean surfaces (1bit/pixel)																																																									
18	PLANAR_422_8	Sample_8x8 AVS only																																																									

MEDIA_SURFACE_STATE		
	Others	Reserved
26	Interleave Chroma	
	Format:	Enable
<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>		
25	Cr(V)/Cb(U) Pixel Offset U Direction	
	Default Value:	0
	Format:	U0.1
	Description	
<p>Specifies the distance to the U/V values with respect to the even numbered Y channels in the U direction</p>		
Programming Notes		
<p>This field must be zero for all formats except PLANAR_420_8, PLANAR_422_8, YCRCB_NORMAL, YCRCB_SWAPUVY, YCRCB_SWAPUV, YCRCB_SWAPY.</p>		
24	Cr(V)/Cb(U) Pixel Offset V Direction MSB	
	Default Value:	0
	Format:	U1
	Description	
<p>Specifies the distance to the U/V values with respect to the even numbered Y channels in the V direction</p>		
Programming Notes		
<p>This field must be zero for all formats except PLANAR_420_8.</p> <p>This offset has been increased from 2 bits to 3 bits as U1.2 format and this bit is used in conjunction with the bits in the Cr(V)/Cb(U) Pixel Offset V Direction field in DWORD 1, which contain the rest of the bits for offset V-direction. Valid values for the combined field range from 0 to 4.</p>		
23	Memory Compression Mode	
	Distinguishes Vertical from Horizontal compression.	
	Value	Name
	Description	
0	Horizontal Compression Mode [Default]	
1	Vertical Compression Mode	

MEDIA_SURFACE_STATE			
22	Memory Compression Enable		
	Format:	Enable	
	This surface may contain compressed or compressible pixels. Memory compression will be attempted for writes to this surface. Reads from this surface will check for compressed data.		
	Programming Notes		
	The compression control must have 0 value for non-tileY modes.		
	Please refer to vol1a Memory Data Formats chapter -- section Media Memory Compression for more details, including format restrictions.		
	21	Address Control	
		Value	Name
		0	CLAMP
		1	MIRROR
20:3	Surface Pitch		
	Format:	U18-1 pitch in Bytes	
	This field specifies the surface pitch in (#Bytes - 1).		
	Value	Description	
	[0,262143]	For other linear surfaces: representing [1B, 256KB]	
	[511, 262143]	For X-tiled surface: representing [512B, 256KB] = [1 tile, 512 tiles]	
	[127, 262143]	For Y-tiled surfaces: representing [128B, 256KB] = [1 tile, 2048 tiles]	
	Programming Notes		
	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. The Surface Pitches of current picture and reference picture should be declared as the identical type in VDI mode with identical Height, Width and Format.		
	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	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.		
	Programming Notes		
Must be Zero as this field is not used.			

MEDIA_SURFACE_STATE																		
1:0	<p>Tile Mode</p> <table border="1"> <tr> <td>Format:</td> <td>U2 Enumerated Type</td> </tr> </table> <p>This field specifies the type of memory tiling (Linear, WMajor, XMajor, or YMajor) employed to tile this surface. See Memory Interface Functions for details on memory tiling and restrictions.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TILEMODE_LINEAR</td> <td>Linear mode (no tiling)</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td>Reserved</td> </tr> <tr> <td>2h</td> <td>TILEMODE_XMAJOR</td> <td>X major tiling</td> </tr> <tr> <td>3h</td> <td>TILEMODE_YMAJOR</td> <td>Y major tiling</td> </tr> </tbody> </table>	Format:	U2 Enumerated Type	Value	Name	Description	0h	TILEMODE_LINEAR	Linear mode (no tiling)	1h	Reserved	Reserved	2h	TILEMODE_XMAJOR	X major tiling	3h	TILEMODE_YMAJOR	Y major tiling
	Format:	U2 Enumerated Type																
	Value	Name	Description															
0h	TILEMODE_LINEAR	Linear mode (no tiling)																
1h	Reserved	Reserved																
2h	TILEMODE_XMAJOR	X major tiling																
3h	TILEMODE_YMAJOR	Y major tiling																
Programming Notes																		
<ul style="list-style-type: none"> Refer to <i>Memory Data Formats</i> for restrictions on TileMode direction for the various buffer types. (Of particular interest is the fact that YMAJOR tiling is not supported for display/overlay buffers). The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this field. Linear surfaces can be mapped to Main Memory (uncached) or System Memory (cacheable, snooped). Tiled (X/Y/W) surfaces can only be mapped to Main Memory. 																		
3	31:30	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ														
	Format:	MBZ																
	29:16	<p>X Offset for U(Cb)</p> <table border="1"> <tr> <td>Format:</td> <td>U14 Pixel Offset</td> </tr> </table> <p style="text-align: center;">Description</p> <p>For non planar surfaces this field specifies the horizontal offset in pixels from the Surface Base Address to the start (origin) of the surface.</p> <p>For Planar surfaces this field specifies the horizontal offset in pixels from the Y-plane origin to the start (origin) of the U(Cb) plane or the interleaved UV plane if Interleave Chroma is enabled. Resultant X-offset = 'X-offset of the surface (Y-plane)' + 'X offset for U(Cb)'</p> <p>For TileYS and TileYF this offset should be integral multiple of Tile width of Luma plane.</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:	U14 Pixel Offset														
Format:	U14 Pixel Offset																	
15:14	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ															
Format:	MBZ																	

MEDIA_SURFACE_STATE											
4	13:0	Y Offset for U(Cb) <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U14 Row Offset</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Description</th> </tr> <tr> <td>For non planar surfaces this field specifies the vertical offset in pixels from the Surface Base Address to the start (origin) of the surface.</td> </tr> <tr> <td>For Planar surfaces this field specifies the vertical offset in rows from the Y-plane origin to the start (origin) of the U(Cb) plane or the interleaved UV plane if Interleave Chroma is enabled. Resultant X-offset = 'Y-offset of the surface (Y-plane)' + 'Y offset for U(Cb)'</td> </tr> <tr> <td>For TileYS and TileYF this offset should be integral multiple of Tile width of Luma plane.</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>This field must be aligned by 4 bit[1:0] = 00</td> </tr> <tr> <td>This field must be aligned by 4 bit[1:0] = 00 for all format besides PLANAR_420_*</td> </tr> </table>	Format:	U14 Row Offset	Description	For non planar surfaces this field specifies the vertical offset in pixels from the Surface Base Address to the start (origin) of the surface.	For Planar surfaces this field specifies the vertical offset in rows from the Y-plane origin to the start (origin) of the U(Cb) plane or the interleaved UV plane if Interleave Chroma is enabled. Resultant X-offset = 'Y-offset of the surface (Y-plane)' + 'Y offset for U(Cb)'	For TileYS and TileYF this offset should be integral multiple of Tile width of Luma plane.	Programming Notes	This field must be aligned by 4 bit[1:0] = 00	This field must be aligned by 4 bit[1:0] = 00 for all format besides PLANAR_420_*
	Format:	U14 Row Offset									
	Description										
	For non planar surfaces this field specifies the vertical offset in pixels from the Surface Base Address to the start (origin) of the surface.										
	For Planar surfaces this field specifies the vertical offset in rows from the Y-plane origin to the start (origin) of the U(Cb) plane or the interleaved UV plane if Interleave Chroma is enabled. Resultant X-offset = 'Y-offset of the surface (Y-plane)' + 'Y offset for U(Cb)'										
	For TileYS and TileYF this offset should be integral multiple of Tile width of Luma plane.										
	Programming Notes										
	This field must be aligned by 4 bit[1:0] = 00										
	This field must be aligned by 4 bit[1:0] = 00 for all format besides PLANAR_420_*										
	31:30	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ							
Format:	MBZ										
29:16	X Offset for V(Cr) <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>///$[(\text{Surface Format}] \text{ is one of planar}) \text{ AND } ([\text{Interleave Chroma}] == '0')$</td> </tr> <tr> <td>Format:</td> <td>U14 Pixel Offset</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Description</th> </tr> <tr> <td>For Planar surfaces this field specifies the horizontal offset in pixels from the Y-plane origin to the start (origin) of the V(Cb) plane. Resultant X-offset = 'X-offset of the surface (Y-plane)' + 'X offset for V(Cb)'</td> </tr> <tr> <td>For TileYS and TileYF this offset should be integral multiple of Tile width of Luma plane.</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.</td> </tr> </table>	Exists If:	/// $[(\text{Surface Format}] \text{ is one of planar}) \text{ AND } ([\text{Interleave Chroma}] == '0')$	Format:	U14 Pixel Offset	Description	For Planar surfaces this field specifies the horizontal offset in pixels from the Y-plane origin to the start (origin) of the V(Cb) plane. Resultant X-offset = 'X-offset of the surface (Y-plane)' + 'X offset for V(Cb)'	For TileYS and TileYF this offset should be integral multiple of Tile width of Luma plane.	Programming Notes	For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.	
Exists If:	/// $[(\text{Surface Format}] \text{ is one of planar}) \text{ AND } ([\text{Interleave Chroma}] == '0')$										
Format:	U14 Pixel Offset										
Description											
For Planar surfaces this field specifies the horizontal offset in pixels from the Y-plane origin to the start (origin) of the V(Cb) plane. Resultant X-offset = 'X-offset of the surface (Y-plane)' + 'X offset for V(Cb)'											
For TileYS and TileYF this offset should be integral multiple of Tile width of Luma plane.											
Programming Notes											
For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.											
15	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										

MEDIA_SURFACE_STATE		
	14:0	Y Offset for V(Cr)
		Exists If: <code>///([Surface Format] is one of planar) AND ([Interleave Chroma] == '0')</code>
		Format: U15 Row Offset
		<p style="text-align: center;">Description</p> <p>For Planar surfaces this field specifies the vertical offset in rows from the Y-plane origin to the start (origin) of the V(Cb) plane. Resultant Y-offset = 'Y-offset of the surface (Y-plane)' + 'Y offset for V(Cb)'</p> <p>For TileYS and TileYF this offset should be integral multiple of Tile width of Luma plane.</p> <p style="text-align: center;">Programming Notes</p> <p>This field must indicate a multiple of 4 (bit 0 & 1 = 00).</p>
5	31	Vertical Line Stride
		Format: U1 in lines to skip between logically adjacent lines
		For Surfaces accessed via the sample_8x8 message: Specifies number of lines (0 or 1) to skip between logically adjacent lines - provides support of interleaved (field) surfaces as textures. For Other Surfaces: Vertical Line Stride must be zero.
	30	Vertical Line Stride Offset
		Format: U1 in lines of initial offset (when Vertical Line Stride == 1)
	For Surfaces accessed via the sample_8x8 message: Specifies the offset of the initial line from the beginning of the buffer. For Other Surfaces: Vertical Line Stride Offset must be zero.	
		Programming Notes
		This field must be set to 0 if Vertical Line Stride is 0.
	29:24	Reserved
		Format: MBZ
	23:20	Reserved
		Format: MBZ

MEDIA_SURFACE_STATE																		
19:18	<p>Tiled Resource 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 Sampling Engine, Render Target, and Typed/Untyped Surfaces: This field specifies the tiled resource mode. For other surfaces: This field is ignored.</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> <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.</p> <p>If this field is not set to TRMODE_NONE, the Surface Format must be one with 8, 16, 32, 64, or 128 bits per element, or one of the compressed texture modes (BC*, ETC*, EAC*, ASTC*). Additionally, YCRCB* formats are supported and treated as 16 bits per element, and the PLANAR_420_8 and PLANAR_422_8 formats are supported and treated as 8 bits per element on the Y plane and 16 bits per element on the UV plane (if Interleave Chroma is enabled) or 8 bits per element on the U and V planes (if Interleave Chroma is disabled).</p>	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																	
17:7	<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																
<p>Surface Memory Object Control State</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Default Value:</td> <td>0h DefaultVaueDesc</td> </tr> <tr> <td>Format:</td> <td>MEMORY_OBJECT_CONTROL_STATE</td> </tr> </table> <p>This 7-bit field is used in various state commands and indirect state objects to define cacheability and other attributes related to memory objects.</p>	Default Value:	0h DefaultVaueDesc	Format:	MEMORY_OBJECT_CONTROL_STATE														
Default Value:	0h DefaultVaueDesc																	
Format:	MEMORY_OBJECT_CONTROL_STATE																	
6	<p>Surface Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:0]</td> </tr> </table> <p>Specifies the low 32 bits of the byte-aligned base address of the surface.</p> <p style="text-align: center;">Programming Notes</p> <p>For SURFTYPE_BUFFER render targets, this field specifies the base address of first element of the surface. The surface is interpreted as a simple array of that single element type. The address must be naturally-aligned to the element size (e.g., a buffer containing R32G32B32A32_FLOAT elements must be 16-byte aligned).For SURFTYPE_BUFFER non-rendertarget surfaces, this field specifies the base address of the first element of the surface, computed in software by adding the surface base address to the byte offset of the element in the buffer.Mipmapped, cube and 3D sampling engine surfaces are stored in a 'monolithic' (fixed) format, and only require a single address for the base texture.Linear render target surface base addresses must be element-size aligned, for non-YUV surface formats, or a multiple of 2 element-sizes for YUV surface formats. Other linear surfaces have no alignment requirements (byte alignment is sufficient.)Linear depth buffer surface base addresses must be 64-byte aligned. Note that while render targets (color) can be SURFTYPE_BUFFER, depth buffers cannot.Tiled surface base</p>	Format:	GraphicsAddress[31:0]															
	Format:	GraphicsAddress[31:0]																

MEDIA_SURFACE_STATE				
		addresses must be 4KB-aligned. Note that only the offsets from Surface Base Address are tiled, Surface Base Address itself is not transformed using the tiling algorithm. For tiled surfaces, the actual start of the surface can be offset from the Surface Base Address by the X Offset and Y Offset fields. Certain message types used to access surfaces have more stringent alignment requirements. Please refer to the specific message documentation for additional restrictions.		
7	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>Surface 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>Specifies the high 16 bits of the byte-aligned base address of the surface. Refer to Surface Base Address [31:0] for programming notes applying to this field.</p>	Format:	GraphicsAddress[47:32]	
Format:	GraphicsAddress[47:32]			

MEMORY_OBJECT_CONTROL_STATE

MEMORY_OBJECT_CONTROL_STATE		
Source:	BSpec	
Size (in bits):	7	
Default Value:	0x00000000	
DWord	Bit	Description
0	6:1	Index to 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

MemoryAddressAttributes

MemoryAddressAttributes			
Source:	BSpec		
Size (in bits):	32		
Default Value:	0x00000000		
<p>This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. It defines the SKL 32-bit memory address attributes for the third DWord of the HCP command buffer address.</p>			
DWord	Bit	Description	
0	31:15	Reserved	
		Format: MBZ	
	14:13	Base Address - Tiled Resource Mode	
		Format: U2	
		For Media Surfaces: This field specifies the tiled resource mode.	
Value		Name	Description
00b		TRMODE_NONE	TileY resources
01b	TRMODE_TILEYF	4KB tiled resources	
10b	TRMODE_TILEYS	64KB tiled resources	
11b	Reserved		
12	Base Address - Row Store Scratch Buffer Cache Select		
	Format: U1		
	This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.		
	Value	Name	Description
0		Buffer going to LLC.	
1		Buffer going to Internal Media Storage.	
11	Reserved		
	Format: MBZ		
8:7	Base Address - Arbitration Priority Control		
	Format: HEVC_ARBITRATION_PRIORITY		
6:1	Base Address - Index to Memory Object Control State (MOCS) Tables		
	Format: U6		
	<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		

Merged Media Block Message Header

MH_MBM - Merged Media Block Message Header				
Source:	DataPort 1			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	X Offset <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S31</td> </tr> </table> <p>X offset (in bytes) of the upper left corner of the block into the surface.</p>	Format:	S31
Format:	S31			
1	31:0	Y Offset <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S31</td> </tr> </table> <p>Y offset (in rows) of the upper left corner of the block into the surface.</p>	Format:	S31
Format:	S31			
2	31:0	Merged Media Block Message Control <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 35%;">Format:</td> <td style="width: 65%;">MHC_MBM_CONTROL</td> </tr> </table> <p>Specifies the Merged message subtype and additional input parameters.</p>	Format:	MHC_MBM_CONTROL
Format:	MHC_MBM_CONTROL			
3	31:0	Mask <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> <p>The Mask is ignored by the Merged Media Block message: all Dwords are always returned on reads, and always enabled to be written on writes.</p>	Format:	U32
Format:	U32			
4	31:0	FFTID <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 45%;">Format:</td> <td style="width: 55%;">MHC_FFTID</td> </tr> </table> <p>Fixed Function Thread ID</p>	Format:	MHC_FFTID
Format:	MHC_FFTID			
5-7	95:0	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

Merged Media Block Message Header Control

MHC_MBM_CONTROL - Merged Media Block Message Header Control											
Source:	BSpec										
Size (in bits):	32										
Default Value:	0x00000000										
DWord	Bit	Description									
0	31:30	Message Mode Format: Enumeration Specifies the Media Block Read message is Normal subtype.									
		<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;">00h</td> <td style="text-align: center;">Normal</td> <td>The Block Height and Block Width fields are specified in this Dword. The Mask is ignored by a media block read message.</td> </tr> <tr> <td style="text-align: center;">Others</td> <td style="text-align: center;">Reserved</td> <td>Reserved.</td> </tr> </tbody> </table>	Value	Name	Description	00h	Normal	The Block Height and Block Width fields are specified in this Dword. The Mask is ignored by a media block read message.	Others	Reserved	Reserved.
		Value	Name	Description							
00h	Normal	The Block Height and Block Width fields are specified in this Dword. The Mask is ignored by a media block read message.									
Others	Reserved	Reserved.									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Ignore</td> </tr> </table> Ignored		Format:	Ignore								
Format:	Ignore										
29	28:24	Reserved Format: Ignore Ignored									
		Sub-Register Offset Format: U5 Provides the sub-register offset in unit of bytes of a Merged Media Block Read message. This field is ignored (reserved) for a media block write message. Range = [0, 28]. Only a multiple of BasePitch, including 0, is valid.									
		Programming Notes									
		Sub-Register Offset and Register Pitch Control allow software to assembly multiple media block reads directly into a shared GRF register set. For example, if both are set to zero, the read data are written to GRF registers, aligning to the least significant bits of the first register, and the register pitch is equal to the next power-of-2 that is greater than or equal to the Block Width. If Register Pitch Control is non-zero, multiple media block read messages sharing the same Register Pitch Control but with different Sub-Register Offset can fill in the same set of GRF registers with media block data line interleaved.									
Restriction		For the Sampler Cache Data, this field must be zero. BasePitch is defined as the next the power-of-2 that is greater than or equal to the Block Width. Minimum BasePitch is 1 DWord. Sub-Register Offset must be aligned to BasePitch (therefore will be a multiple of DWords as well). When Register Pitch Control = 0, Sub-Register Offset must align to BasePitch*Block Height. ensuring the output fits in a single GRF register. In general (and specifically when Sub-Register Offset is greater than 0), when the resulting data will cross a GRF register boundary, the data must be placed symmetrically between GRF registers.									

Message Descriptor - Render Target Write

Message Descriptor - Render Target Write											
Source:	BSpec										
Size (in bits):	32										
Default Value:	0x00000000										
DWord	Bit	Description									
0	31	Reserved Format: MBZ									
	30	Data Format Format: U1 <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Single Precision</td> <td>32b</td> </tr> <tr> <td>1</td> <td>Half Precision</td> <td>16b</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> This field is applicable for Render Target Write Messages ONLY.	Value	Name	Description	0	Single Precision	32b	1	Half Precision	16b
	Value	Name	Description								
	0	Single Precision	32b								
	1	Half Precision	16b								
29:14	Reserved Format: MBZ										
13	Per-Sample PS outputs enable This bit must not be set when Render Target is not bound to pixel-shader OR when Render Target is not multisampled. This bit must be set when PS runs at sample-frequency i.e. pixel shader dispatch mode is PER_SAMPLE. By setting this bit, PS sends Render Target Write Message that outputs color, depth(optional) and stencil(optional) phases on per sample basis for each slot. When Render Target is multisampled and this bit is reset, Render Target outputs color, depth(optional) and stencil(optional) at pixel frequency. It should be noted that the latter case is applicable for only per-pixel PS invocation.										
12	Last Render Target Select 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. <p style="text-align: center;">Programming Notes</p> In general, when threads are not launched by 3D FF, this bit must be zero.										
11	Slot Group Select This field selects whether slots 15:0 or slots 31:16 are used for bypassed data. Bypassed data includes the antialias alpha, multisample coverage mask, and if the header is not present also includes the X/Y addresses and pixel enables. For 8- and 16-pixel dispatches, SLOTGRP_LO must be selected on every message. For 32-pixel dispatches, this field must be set correctly for each message based on which slots are currently being processed.										

Message Descriptor - Render Target Write																							
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SLOTGRP_LO</td> <td>choose bypassed data for slots 15:0</td> </tr> <tr> <td>1</td> <td>SLOTGRP_HI</td> <td>choose bypassed data for slots 31:16</td> </tr> </tbody> </table>	Value	Name	Description	0	SLOTGRP_LO	choose bypassed data for slots 15:0	1	SLOTGRP_HI	choose bypassed data for slots 31:16													
Value	Name	Description																					
0	SLOTGRP_LO	choose bypassed data for slots 15:0																					
1	SLOTGRP_HI	choose bypassed data for slots 31:16																					
	Programming Notes																						
	For SIMD8 Image Write message thsi field MBZ.																						
10:8	<p>Message Type This field specifies the type of render target message. For the SIMD8_DUALSRC_xx messages, the low bit indicates which slots to use for the pixel enables, X/Y addresses, and oMask.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>SIMD16</td> <td>SIMD16 single source message</td> </tr> <tr> <td>001b</td> <td>SIMD16_REPDATA</td> <td>SIMD16 single source message with replicated data</td> </tr> <tr> <td>010b</td> <td>SIMD8_DUALSRC_LO</td> <td>SIMD8 dual source message, use slots 7:0</td> </tr> <tr> <td>011b</td> <td>SIMD8_DUALSRC_HI</td> <td>SIMD8 dual source message, use slots 15:8</td> </tr> <tr> <td>100b</td> <td>SIMD8_LO</td> <td>SIMD8 single source message, use slots 7:0</td> </tr> <tr> <td>111b</td> <td>SIMD16_REPDATA</td> <td>It's only supported when accessing <i>Tiled Memory</i>. Using this Message Type to access linear (<i>Untiled</i>) memory is UNDEFINED.</td> </tr> </tbody> </table>		Value	Name	Description	000b	SIMD16	SIMD16 single source message	001b	SIMD16_REPDATA	SIMD16 single source message with replicated data	010b	SIMD8_DUALSRC_LO	SIMD8 dual source message, use slots 7:0	011b	SIMD8_DUALSRC_HI	SIMD8 dual source message, use slots 15:8	100b	SIMD8_LO	SIMD8 single source message, use slots 7:0	111b	SIMD16_REPDATA	It's only supported when accessing <i>Tiled Memory</i> . Using this Message Type to access linear (<i>Untiled</i>) memory is UNDEFINED.
Value	Name	Description																					
000b	SIMD16	SIMD16 single source message																					
001b	SIMD16_REPDATA	SIMD16 single source message with replicated data																					
010b	SIMD8_DUALSRC_LO	SIMD8 dual source message, use slots 7:0																					
011b	SIMD8_DUALSRC_HI	SIMD8 dual source message, use slots 15:8																					
100b	SIMD8_LO	SIMD8 single source message, use slots 7:0																					
111b	SIMD16_REPDATA	It's only supported when accessing <i>Tiled Memory</i> . Using this Message Type to access linear (<i>Untiled</i>) memory is UNDEFINED.																					
	Programming Notes																						
	the above slots indicated are within the 16 slots selected by Slot Group Select . If SLOTGRP_HI is selected, the SIMD8 message types above reference slots 23:16 or 31:24 instead of 7:0 or 15:8, respectively.																						
	SIMD16_REPDATA message must not be used in SIMD8 pixel-shaders.																						
7:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ																			
Format:	MBZ																						

Message Descriptor - Sampling Engine

Message Descriptor - Sampling Engine				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31	EOT		
	30	Return Format		
		Format:	U1	
		Value	Name	Description
		0	32-bit	Return data is 32b
		1	16-bit	Return data is 16b
	Programming Notes			
	This field must be set to 32-bit for messages with SIMD Mode of SIMD4x2 or SIMD32/64. This field must be set to 32 for resinfo, LOD and sampleinfo messages.			
	This message must be set to 32-bit for ld_mcs			
	29	Reserved	Format: MBZ	
28:25	Message Length	Format: U4		
	This field specifies the number of 256-bit GRF registers starting from (src) to be sent out on the request message payload.			
	Value	Name		
	[1,15]			
Programming Notes				
A value of 0 is considered erroneous.				
24:20	Response Length	Format: U5		
	This field indicates the number of 256-bit registers expected in the message response.			
	Value	Name		
	[0,16]			
Programming Notes				
A value 0 indicates that the request message does not expect any response. The largest response supported is 16 GRF registers.				

Message Descriptor - Sampling Engine							
19	<p>Header Present</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p style="text-align: center;">Description</p> <p>Specifies whether the message includes a header phase. If the header is not present (this field is zero), all of the fields normally contained in the header are assumed to be 0.</p> <p>If the header is not present, in some cases the Write Channel Mask fields are set according to the Response Length.</p>	Format:	Enable				
Format:	Enable						
18:17	<p>SIMD Mode[1:0]</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2</td> </tr> </table> <p>Specifies the SIMD mode of the message being sent.</p> <p style="text-align: center;">Programming Notes</p> <p>The SIMD Mode Extension field in the message header distinguishes between SIMD8D and SIMD4x2 modes.</p>	Format:	U2				
Format:	U2						
16:12	<p>Message Type</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U5</td> </tr> </table> <p>Specifies the type of message being sent. For more details, please refer to Message Format section for the definition of these 5 bits..</p>	Format:	U5				
Format:	U5						
11:8	<p>Sampler Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U4</td> </tr> </table> <p>Specifies the index into the sampler state table. Ignored for Id, resinfo, sampleinfo, and cache_flush type messages.</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>[0,15]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> For the deinterlace message, this field must be a multiple of 2 (even). For the sample_8x8 message, this field must be a multiple of 4. 	Format:	U4	Value	Name	[0,15]	
Format:	U4						
Value	Name						
[0,15]							
7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U8</td> </tr> </table> <p>Specifies the index into the binding table. Ignored for cache_flush type messages. Values of 255 and 253 indicate stateless. 254 indicates SLM. 252 indicates bindless.</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>[0,255]</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	[0,255]	
Format:	U8						
Value	Name						
[0,255]							

MFD_MPEG2_BSD_OBJECT Inline Data Description

MFD_MPEG2_BSD_OBJECT Inline Data Description											
Source:	VideoCS										
Size (in bits):	64										
Default Value:	0x00000000, 0x00000000										
DW0..1 corresponds to DW3..4 of the MFD_MPEG2_BSD_OBJECT.											
DWord	Bit	Description									
0	31:24	<p>Slice Horizontal Position</p> <table border="1"> <tr> <td>Format:</td> <td>U8 in Macroblocks</td> </tr> </table> <p>This field indicates the horizontal position of the first macroblock in the slice.</p>	Format:	U8 in Macroblocks							
	Format:	U8 in Macroblocks									
	23:16	<p>Slice Vertical Position</p> <table border="1"> <tr> <td>Format:</td> <td>U8 in Macroblocks</td> </tr> </table> <p>This field indicates the vertical position of the first macroblock in the slice.</p>	Format:	U8 in Macroblocks							
	Format:	U8 in Macroblocks									
15:8	<p>Macroblock Count</p> <table border="1"> <tr> <td>Format:</td> <td>U8 in Macroblocks</td> </tr> </table> <p>This field indicates the number of macroblocks in the slice, including skipped macroblocks.</p>	Format:	U8 in Macroblocks								
Format:	U8 in Macroblocks										
7	<p>Slice Concealment Override Bit</p> <p>This bit forces hardware to handle the current slice in Conceal or Deocode Mode. If this bit is set to one, VIN will force the current slice to do concealment or to decode from bitstream regardless if the slice boundary has errors or not.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td></td> <td>VIN will use driver-provided "Slice Concealment Type" regardless of valid slice boundary</td> </tr> <tr> <td>0h</td> <td></td> <td>Driver must program "Slice Concealment Type" to '0'. VIN will set "Slice Concealment Type" depending if the slice boundary has error or not</td> </tr> </tbody> </table>	Value	Name	Description	1h		VIN will use driver-provided "Slice Concealment Type" regardless of valid slice boundary	0h		Driver must program "Slice Concealment Type" to '0'. VIN will set "Slice Concealment Type" depending if the slice boundary has error or not	
Value	Name	Description									
1h		VIN will use driver-provided "Slice Concealment Type" regardless of valid slice boundary									
0h		Driver must program "Slice Concealment Type" to '0'. VIN will set "Slice Concealment Type" depending if the slice boundary has error or not									
6		<p>Slice Concealment Type Bit</p> <p>This bit can be forced by driver ("Slice Concealment Override Bit") or set by VINunit depending on slice boundary errors.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td></td> <td>VMD will conceal all MBs of the slice regardless of bitstream. (If driver does not force the value of this bit, VIN will set this bit depending on slice boundary error. If the next slice position of the current slice is out-of-bound or the same or earlier than the current slice start position, VIN will set this bit for the next slice)</td> </tr> <tr> <td>0h</td> <td></td> <td>VMD will decode MBs from the bitstream until the bitstream is run-out. Then VMD will conceal the remaining MBs.</td> </tr> </tbody> </table>	Value	Name	Description	1h		VMD will conceal all MBs of the slice regardless of bitstream. (If driver does not force the value of this bit, VIN will set this bit depending on slice boundary error. If the next slice position of the current slice is out-of-bound or the same or earlier than the current slice start position, VIN will set this bit for the next slice)	0h		VMD will decode MBs from the bitstream until the bitstream is run-out. Then VMD will conceal the remaining MBs.
		Value	Name	Description							
		1h		VMD will conceal all MBs of the slice regardless of bitstream. (If driver does not force the value of this bit, VIN will set this bit depending on slice boundary error. If the next slice position of the current slice is out-of-bound or the same or earlier than the current slice start position, VIN will set this bit for the next slice)							
	0h		VMD will decode MBs from the bitstream until the bitstream is run-out. Then VMD will conceal the remaining MBs.								
Programming Notes											
VIN can turn this bit from 0 to 1 internally if "Slice Concealment Disable Bit" is "0" and VIN detects slice boundary errors.											

MFD_MPEG2_BSD_OBJECT Inline Data Description											
1	5	<p>Last Pic Slice This bit is added to support error concealment at the end of a picture.</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>1h</td> <td></td> <td>The current Slice is the last Slice of the entire picture</td> </tr> <tr> <td>0h</td> <td></td> <td>The current Slice is not the last Slice of current picture</td> </tr> </tbody> </table>	Value	Name	Description	1h		The current Slice is the last Slice of the entire picture	0h		The current Slice is not the last Slice of current picture
	Value	Name	Description								
	1h		The current Slice is the last Slice of the entire picture								
	0h		The current Slice is not the last Slice of current picture								
	4	Reserved									
	3	<p>Is Last MB</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>1h</td> <td></td> <td>The current MB is the last MB in the current Slice</td> </tr> <tr> <td>0h</td> <td></td> <td>The current MB is not the last MB in the current Slice</td> </tr> </tbody> </table>	Value	Name	Description	1h		The current MB is the last MB in the current Slice	0h		The current MB is not the last MB in the current Slice
	Value	Name	Description								
	1h		The current MB is the last MB in the current Slice								
	0h		The current MB is not the last MB in the current Slice								
	2:0	<p>First Macroblock Bit Offset</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 provides the bit offset of the first macroblock in the first byte of the input bitstream.</p>	Format:	U3							
Format:	U3										
31:29	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
28:24	<p>Quantizer Scale Code</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U5</td> </tr> </table> <p>This field sets the quantizer scale code of the inverse quantizer. It remains in effect until changed by a decoded quantizer scale code in a macroblock. This field is decoded from the slice header by host software.</p>	Format:	U5								
Format:	U5										
23:17	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
16:8	<p>Next Slice Vertical Position</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">U9 in macroblocks</td> </tr> </table> <p>This field indicates the vertical position (in macroblock units) of the first macroblock in the next slice.</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 style="padding: 5px;">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 the picture (field picture will be in height of field) (since y-direction is zero-based numbering).</td> </tr> </tbody> </table>	Format:	U9 in macroblocks	Programming Notes	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 the picture (field picture will be in height of field) (since y-direction is zero-based numbering).						
Format:	U9 in macroblocks										
Programming Notes											
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 the picture (field picture will be in height of field) (since y-direction is zero-based numbering).											
7:0	<p>Next Slice Horizontal Position</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">U8 in macroblocks</td> </tr> </table> <p>This field indicates the horizontal position (in macroblock units) of the first macroblock in the next slice.</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 style="padding: 5px;">This field is primarily used for error concealment. In the case that current slice is the last slice, this field should set 0.</td> </tr> </tbody> </table>	Format:	U8 in macroblocks	Programming Notes	This field is primarily used for error concealment. In the case that current slice is the last slice, this field should set 0.						
Format:	U8 in macroblocks										
Programming Notes											
This field is primarily used for error concealment. In the case that current slice is the last slice, this field should set 0.											

MPEG2

MPEG2				
Source:	VideoCS			
Size (in bits):	16			
Default Value:	0x00000000			
DWord	Bit	Description		
0	15:6	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		
	5	Missing EOB Error This flag indicates missing EOB SEs coded in the bit-stream. Missing EOBs are concealed to match CBP of the error MB.		
	4	Inconsistent starting position Error - overlapping MBs This flag indicates two slices overlapping one another by one or more MBs. Duplicate MBs decoded off the second slice shall be discarded.		
	3	Slice out-of-bound Error This flag indicates a slice is running beyond the width of the picture. Out-of-bound MBs shall be discarded.		
	2	Premature frame end Error This flag indicates missing slices/MBs coded in the bit-stream of a frame. One or more MBs are concealed to reach end of picture.		
	1	Inconsistent starting position Error - Missing MBs This flag indicates one or more MBs are being concealed due to inconsistent MB starting and ending positions between slices.		
	0	MB Concealment Flag . Each pulse from this flag indicates one MB is concealed by hardware.		

MPEG4-2_Inline_DMEM		
6	31:1	Reserved Format: MBZ
	0	hw_interlaced MPEG4-2 SE: interlaced
7	31:1	Reserved Format: MBZ
	0	hw_obmc_disable MPEG4-2 SE: obmc_disable <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> OBMC currently not supported; must be set to zero
8	31:1	Reserved Format: MBZ
	0	hw_sprite_enable MPEG4-2 SE: sprint_enable
9	31:6	Reserved Format: MBZ
	5:0	hw_sprite_warping_points MPEG4-2 SE: no_of_sprite_warping_points <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> Only 0 and 1 supported
10	31:2	Reserved Format: MBZ
	1:0	hw_sprite_warping_accuracy MPEG4-2 SE: sprite_warping_accuracy <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> Only 1/2, 1/4, 1/8 supported
11	31:4	Reserved Format: MBZ
	3:0	hw_quant_precision MPEG4-2 SE: quant_precision <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> Must be set to 5h for ASP/SP profiles
12	31:1	Reserved Format: MBZ
	0	hw_quant_type MPEG4-2 SE: quant_type

MPEG4-2_Inline_DMEM		
13	31:1	Reserved Format: MBZ
	0	hw_quarter_sample MPEG4-2 SE: quarter_sample
14	31:1	Reserved Format: MBZ
	0	hw_resync_marker_disable MPEG4-2 SE: resync_marker_disable
15	31:1	Reserved Format: MBZ
	0	hw_data_partitioned MPEG4-2 SE: data_partitioned
	Programming Notes Data partitioning currently not supported; must be set to zero	
16	31:1	Reserved Format: MBZ
	0	hw_reversible_vlc MPEG4-2 SE: reversible_vlc
17	31:7	Reserved Format: MBZ
	6:0	hw_MacroBlockPerRow Number of macroblocks per row, $\text{trunc}(\text{vop_width} + 15 \gg 4)$
18	31:7	Reserved Format: MBZ
	6:0	hw_MacroBlockPerCol Number of macroblocks per column, $\text{trunc}(\text{vop_height} + 15 \gg 4)$
19	31:15	Reserved Format: MBZ
	14:0	hw_MacroBlockPerVOP Number of macroblocks per VOP, $\text{MacroBlockPerRow} * \text{MacroBlockPerCol}$
20	31:4	Reserved Format: MBZ
	3:0	hw_length_of_MB_number_code Length of macroblock number code (1-14) in Table 6-27 column one of the MPEG4-2 standard specification
21	31:0	hw_Tframe Tframe calculation is described in section 7.7.2.2 Motion vector decoding in B-VOP in the MPEG4-2 standard specification

MPEG4-2_Inline_DMEM			
22	31:0	hw_TRD TRD calculation is described in section 7.7.2.2 Motion vector decoding in B-VOP in the MPEG4-2 standard specification	
23	31:0	hw_TRB TRB calculation is described in section 7.7.2.2 Motion vector decoding in B-VOP in the MPEG4-2 standard specification	
24	31:2	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td></tr></table> MBZ	
1:0	hw_coding_type MPEG4-2 SE: vop_coding_type		
25	31:1	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td></tr></table> MBZ	
0	hw_rounding_type MPEG4-2 SE: rounding_type		
26	31:3	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td></tr></table> MBZ	
2:0	hw_intra_dc_vlc_thr MPEG4-2 SE: intra_dc_vlc_thr		
27	31:1	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td></tr></table> MBZ	
0	hw_top_field_first MPEG4-2 SE: top_field_first		
28	31:1	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td></tr></table> MBZ	
0	hw_alt_vertical_scan_flag MPEG4-2 SE: alt_vertical_scan_flag		
29	31:0	hw_warping_mv_code_du[0] MPEG4-2 SE: warping_mv_code(du[0])	
30	31:0	hw_warping_mv_code_du[1] MPEG4-2 SE: warping_mv_code(du[1])	
31	31:0	hw_warping_mv_code_du[2] MPEG4-2 SE: warping_mv_code(du[2])	
32	31:0	hw_warping_mv_code_du[3] MPEG4-2 SE: warping_mv_code(du[3])	
33	31:0	hw_warping_mv_code_dv[0] MPEG4-2 SE: warping_mv_code(dv[0])	
34	31:0	hw_warping_mv_code_dv[1] MPEG4-2 SE: warping_mv_code(dv[1])	
35	31:0	hw_warping_mv_code_dv[2] MPEG4-2 SE: warping_mv_code(dv[2])	

MPEG4-2_Inline_DMEM		
36	31:0	hw_warping_mv_code_dv[3] MPEG4-2 SE: warping_mv_code(dv[3])
37	31:5	Reserved Format: MBZ
	4:0	hw_quant MPEG4-2 SE: vop_quant for non-short header mode only (see hw_263_vop_quant for short header mode)
38	31:3	Reserved Format: MBZ
	2:0	hw_fcode_forward MPEG4-2 SE: vop_fcode_forward
39	31:3	Reserved Format: MBZ
	2:0	hw_fcode_backward MPEG4-2 SE: vop_fcode_backward
40	31:9	Reserved Format: MBZ
	8:0	hw_quant_scale MPEG4-2 SE: quant_scale
41	31:8	Reserved Format: MBZ
	7:0	hw_263_temporal_reference MPEG4-2 SE: temporal_reference (short header format only)
42	31:5	Reserved Format: MBZ
	4:0	hw_263_vop_quant MPEG4-2 SE: vop_quant (short header format only)
43	31:5	Reserved Format: MBZ
	4:0	hw_263_gob_number MPEG4-2 SE: gob_number (short header format only)
44	31:0	hw_263_num_gobs_in_vop Derived from Table 6-29 in the MPEG4-2 standard specification (short header format only)
45	31:0	hw_263_num_macroblocks_in_gob Derived from Table 6-29 in the MPEG4-2 standard specification (short header format only)
46	31:1	Reserved Format: MBZ

MPEG4-2_Inline_DMEM				
	0	hw_263_gob_header_empty MPEG4-2 SE: gob_header_empty (short header format only)		
47	31:1	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		
0	hw_263_gob_frame_id MPEG4-2 SE: gob_frame_id (short header format only)			
48	31:9	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		
8:0	hw_263_quant_scale MPEG4-2 SE: quant_scale (short header format only)			
49	31:0	hw_263_num_rows_in_gob Refer to Table 6-29 in the MPEG4-2 standard specification, vop_height/(16*num_gobs_in_vop) (short header format only)		

MSAA Sample Number Message Address Control

MACD_MSAA_SN - MSAA Sample Number Message Address Control				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:4	<p>Reserved</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ
	Format:	MBZ		
3:0	<p>Sample Number</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies the sample number for the slot. If the sample number is larger than the Number of Multisamples in the Surface State, then the access is out of bounds.</p>	Format:	U4	
Format:	U4			

MsgDescpt31

MsgDescpt31						
Source:	Eulsa					
Size (in bits):	29					
Default Value:	0x00000000					
DWord	Bit	Description				
0	28:25	<p>Message Length</p> <p>This field specifies the number of 256-bit MRF registers starting from <curr_dest> to be sent out on the request message payload. Valid value ranges from 1 to 15. A value of 0 is considered erroneous.</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;">1-15</td> <td>Number of MRF Registers</td> </tr> </tbody> </table>	Value	Name	1-15	Number of MRF Registers
	Value	Name				
	1-15	Number of MRF Registers				
	24:20	<p>Response Length</p> <p>This field indicates the number of 256-bit registers expected in the message response. The valid value ranges from 0 to 16. A value 0 indicates that the request message does not expect any response. The largest response supported is 16 GRF registers.</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;">0-16</td> <td>Number of Registers</td> </tr> </tbody> </table>	Value	Name	0-16	Number of Registers
Value	Name					
0-16	Number of Registers					
19	<p>Header Present</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;">Enable</td> </tr> </table> <p>If set, indicates that the message includes a header. Depending on the target shared function, this field may be restricted to either enabled or disabled. Refer to the specific shared function section for details.</p>	Format:	Enable			
Format:	Enable					
18:0	<p>Function Control</p> <p>This field is intended to control the target function unit. Refer to the section on the specific target function unit for details on the contents of this field.</p>					

Normal Media Block Message Header

MH_MB - Normal Media Block Message Header						
Source:	DataPort 1					
Size (in bits):	256					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000					
DWord	Bit	Description				
0	31:0	<p>X Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">S31</td> </tr> </table> <p>X offset (in bytes) of the upper left corner of the block into the surface.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table> <p>Must be DWord aligned (Bits 1:0 MBZ) for the write form of the message.</p>	Format:	S31	Programming Notes	
Format:	S31					
Programming Notes						
1	31:0	<p>Y Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">S31</td> </tr> </table> <p>Y offset (in rows) of the upper left corner of the block into the surface.</p>	Format:	S31		
Format:	S31					
2	31:0	<p>Normal Media Block Message Control</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">MHC_MB_CONTROL</td> </tr> </table> <p>Specifies the Normal message subtype and additional input parameters.</p>	Format:	MHC_MB_CONTROL		
Format:	MHC_MB_CONTROL					
3	31:0	<p>Mask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U32</td> </tr> </table> <p>The Mask is ignored by the Normal Media Block message: all Dwords are always returned on reads, and always enabled to be written on writes.</p>	Format:	U32		
Format:	U32					
4	31:0	<p>FFTID</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MHC_FFTID</td> </tr> </table> <p>Fixed Function Thread ID</p>	Format:	MHC_FFTID		
Format:	MHC_FFTID					
5-7	95:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore		
Format:	Ignore					

Normal Media Block Message Header Control

MHC_MB_CONTROL - Normal Media Block Message Header Control											
Source:	BSpec										
Size (in bits):	32										
Default Value:	0x00000000										
DWord	Bit	Description									
0	31:30	Message Mode									
		Format: Enumeration									
		Specifies the interpretation of M0.3 (Pixel or Byte Mask). For the Sampler Cache Data Port, this field is ignored, behaving as if always set to NORMAL.									
		<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;">00h</td> <td style="text-align: center;">Normal</td> <td>The Block Height and Block Width fields are specified in this Dword. The Mask is ignored by a media block read message and behaves as if it is set to all ones for a media block write message.</td> </tr> <tr> <td style="text-align: center;">Others</td> <td style="text-align: center;">Reserved</td> <td>Reserved.</td> </tr> </tbody> </table>	Value	Name	Description	00h	Normal	The Block Height and Block Width fields are specified in this Dword. The Mask is ignored by a media block read message and behaves as if it is set to all ones for a media block write message.	Others	Reserved	Reserved.
		Value	Name	Description							
00h	Normal	The Block Height and Block Width fields are specified in this Dword. The Mask is ignored by a media block read message and behaves as if it is set to all ones for a media block write message.									
Others	Reserved	Reserved.									
Programming Notes											
The Media Block Read message is Normal subtype when both Sub-Register Offset and Register Pitch Control are zero. The Media Block Read message is Merged subtype when either Sub-Register Offset or Register Pitch Control are non-zero.											
29		Reserved									
		Format: Ignore Ignored									
28:24		Sub-Register Offset									
		Format: MBZ The sub-register offset must be 0 for Normal Media Block Read message subtype. This field is ignored (reserved) for a media block write message.									
23:22		Reserved									
		Format: Ignore Ignored									
21:16		Block Height									
		Format: U6 Height in rows of block being accessed. Range = [0,63] representing 1 to 64 rows									
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Restriction</th> </tr> </thead> <tbody> <tr> <td>If Block Width (bytes), then Maximum Block Height (rows) is constrained by (# Dwords width) * (# rows) <= 64 Dwords.</td> </tr> </tbody> </table>	Restriction	If Block Width (bytes), then Maximum Block Height (rows) is constrained by (# Dwords width) * (# rows) <= 64 Dwords.							
Restriction											
If Block Width (bytes), then Maximum Block Height (rows) is constrained by (# Dwords width) * (# rows) <= 64 Dwords.											

MHC_MB_CONTROL - Normal Media Block Message Header Control

	15:10	Reserved	Format: Ignore
			Ignored
	9:8	Register Pitch Control	Format: MBZ
			The register pitch must be 0 for a Normal Media Block Read message. This field is ignored (reserved) for a media block write message.
	7:6	Reserved	Format: Ignore
			Ignored
	5:0	Block Width	Format: U6
			Width in bytes of the block being accessed. For normal Media Block Writes, Range = [0,63] representing 1 to 64 Bytes. For normal Media Block Reads and for masked and merged Media Block messages, Range = [0,31] representing 1 to 32 Bytes.
			Programming Notes
			Must be DWord aligned for the write form of the message.

oMask Message Data Payload Register

MDPR_OMASK - oMask Message Data Payload Register				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	<p>oMask1</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 1. Not used for Slot Group HI.</p>	Format:	U16
	Format:	U16		
15:0	<p>oMask0</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 0. Not used for Slot Group HI.</p>	Format:	U16	
Format:	U16			
1	31:16	<p>oMask3</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 3. Not used for Slot Group HI.</p>	Format:	U16
	Format:	U16		
15:0	<p>oMask2</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 2. Not used for Slot Group HI.</p>	Format:	U16	
Format:	U16			
2	31:16	<p>oMask5</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 5. Not used for Slot Group HI.</p>	Format:	U16
	Format:	U16		
15:0	<p>oMask4</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 4. Not used for Slot Group HI.</p>	Format:	U16	
Format:	U16			
3	31:16	<p>oMask7</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 7. Not used for Slot Group HI.</p>	Format:	U16
	Format:	U16		
15:0	<p>oMask6</p> <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 6. Not used for Slot Group HI.</p>	Format:	U16	
Format:	U16			

MDPR_OMASK - oMask Message Data Payload Register				
4	31:16	<p>oMask9</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 9. Used only if Slot Group HI or SIMD16.</p>	Format:	U16
	Format:	U16		
15:0	<p>oMask8</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 8. Used only if Slot Group HI or SIMD16.</p>	Format:	U16	
Format:	U16			
5	31:16	<p>oMask11</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 11. Used only if Slot Group HI or SIMD16.</p>	Format:	U16
	Format:	U16		
15:0	<p>oMask10</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 10. Used only if Slot Group HI or SIMD16.</p>	Format:	U16	
Format:	U16			
6	31:16	<p>oMask13</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 13. Used only if Slot Group HI or SIMD16.</p>	Format:	U16
	Format:	U16		
15:0	<p>oMask12</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 12. Used only if Slot Group HI or SIMD16.</p>	Format:	U16	
Format:	U16			
7	31:16	<p>oMask15</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 15. Used only if Slot Group HI or SIMD16.</p>	Format:	U16
	Format:	U16		
15:0	<p>oMask14</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>oMask for Pixels [15:0] of Slot 14. Used only if Slot Group HI or SIMD16.</p>	Format:	U16	
Format:	U16			

OM Replicated SIMD16 Render Target Data Payload

MDP_RTW_M16REP - OM Replicated SIMD16 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	oMask
		<table border="1"> <tr> <td>Format:</td> <td>MDPR_OMASK</td> </tr> </table> Slots [15:0] oMask
Format:	MDPR_OMASK	
1.0-1.7	255:0	RGBA
		<table border="1"> <tr> <td>Format:</td> <td>MDPR_RGBA</td> </tr> </table> RGBA for all slots [15:0]
Format:	MDPR_RGBA	

MDP_RTW_MA16 - OM S0A SIMD16 Render Target Data Payload				
5.0-5.7	255:0	Green[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Green	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
6.0-6.7	255:0	Green[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Green	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
7.0-7.7	255:0	Blue[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Blue	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
8.0-8.7	255:0	Blue[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Blue	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
9.0-9.7	255:0	Alpha[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Alpha	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
10.0-10.7	255:0	Alpha[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Alpha	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			

MDP_RTW_M8DS - OM SIMD8 Dual Source Render Target Data Payload				
5.0-5.7	255:0	Src1 Red <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDP_DW_SIMD8</td> </tr> </table> Slots[7:0] or [15:8] of Src1 Red	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
6.0-6.7	255:0	Src1 Green <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDP_DW_SIMD8</td> </tr> </table> Slots[7:0] or [15:8] of Src1 Green	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
7.0-7.7	255:0	Src1 Blue <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDP_DW_SIMD8</td> </tr> </table> Slots[7:0] or [15:8] of Src1 Blue	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
8.0-8.7	255:0	Src1 Alpha <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDP_DW_SIMD8</td> </tr> </table> Slots[7:0] or [15:8] of Src1 Alpha	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			

OM SIMD8 Render Target Data Payload

MDP_RTW_M8 - OM SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	oMask Format: MDPR_OMASK Slots [7:0] oMask. Upper half ignored.
1.0-1.7	255:0	Red Format: MDP_DW_SIMD8 Slots [7:0] Red
2.0-2.7	255:0	Green Format: MDP_DW_SIMD8 Slots [7:0] Green
3.0-3.7	255:0	Blue Format: MDP_DW_SIMD8 Slots [7:0] Blue
4.0-4.7	255:0	Alpha Format: MDP_DW_SIMD8 Slots [7:0] Alpha

MDP_RTW_M16 - OM SIMD16 Render Target Data Payload		
6.0-6.7	255:0	Blue[15:8]
		Format: MDP_DW_SIMD8 Slots [15:8] Blue
7.0-7.7	255:0	Alpha[7:0]
		Format: MDP_DW_SIMD8 Slots [7:0] Alpha
8.0-8.7	255:0	Alpha[15:8]
		Format: MDP_DW_SIMD8 Slots [15:8] Alpha

OS OM S0A SIMD8 Render Target Data Payload

MDP_RTW_SMA8 - OS OM S0A SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1792	
Default Value:	0x00000000, 0x00000000,	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha Format: MDP_DW_SIMD8 Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	oMask Format: MDPR_OMASK Slots [7:0] oMask. Upper half ignored.
2.0-2.7	255:0	Red Format: MDP_DW_SIMD8 Slots [7:0] Red
3.0-3.7	255:0	Green Format: MDP_DW_SIMD8 Slots [7:0] Green
4.0-4.7	255:0	Blue Format: MDP_DW_SIMD8 Slots [7:0] Blue
5.0-5.7	255:0	Alpha Format: MDP_DW_SIMD8 Slots [7:0] Alpha

MDP_RTW_SMA8 - OS OM S0A SIMD8 Render Target Data Payload		
6.0-6.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] Stencil

MDP_RTW_SM8DS - OS OM SIMD8 Dual Source Render Target Data Payload		
5.0-5.7	255:0	Src1 Red Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Red
6.0-6.7	255:0	Src1 Green Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Green
7.0-7.7	255:0	Src1 Blue Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Blue
8.0-8.7	255:0	Src1 Alpha Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Alpha
9.0-9.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] or [15:8] of Stencil

OS S0A SIMD8 Render Target Data Payload

MDP_RTW_SA8 - OS S0A SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha Format: MDP_DW_SIMD8 Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	Red Format: MDP_DW_SIMD8 Slots [7:0] Red
2.0-2.7	255:0	Green Format: MDP_DW_SIMD8 Slots [7:0] Green
3.0-3.7	255:0	Blue Format: MDP_DW_SIMD8 Slots [7:0] Blue
4.0-4.7	255:0	Alpha Format: MDP_DW_SIMD8 Slots [7:0] Alpha
5.0-5.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] Stencil

MDP_RTW_S8DS - OS SIMD8 Dual Source Render Target Data Payload		
		Slots[7:0] or [15:8] of Src1 Green
6.0-6.7	255:0	Src1 Blue Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Blue
7.0-7.7	255:0	Src1 Alpha Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Alpha
8.0-8.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] or [15:8] of Stencil



MDP_RTW_SZMA8 - OS SZ OM S0A SIMD8 Render Target Data Payload		
6.0-6.7	255:0	Source Depth
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Source Depth
Format:	MDP_DW_SIMD8	
7.0-7.7	255:0	Stencil
		<table border="1"> <tr> <td>Format:</td> <td>MDPR_STENCIL</td> </tr> </table> Slots [7:0] Stencil
Format:	MDPR_STENCIL	

MDP_RTW_SZM8DS - OS SZ OM SIMD8 Dual Source Render Target Data Payload

5.0-5.7	255:0	Src1 Red
		Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Red
6.0-6.7	255:0	Src1 Green
		Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Green
7.0-7.7	255:0	Src1 Blue
		Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Blue
8.0-8.7	255:0	Src1 Alpha
		Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Alpha
9.0-9.7	255:0	Source Depth
		Format: MDP_DW_SIMD8 Slots [7:0] or [15:8] of Source Depth
10.0-10.7	255:0	Stencil
		Format: MDPR_STENCIL Slots [7:0] or [15:8] of Stencil

MDP_RTW_SZ8DS - OS SZ SIMD8 Dual Source Render Target Data Payload		
6.0-6.7	255:0	Src1 Blue
		Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Blue
7.0-7.7	255:0	Src1 Alpha
		Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Alpha
8.0-8.7	255:0	Source Depth
		Format: MDP_DW_SIMD8 Slots [7:0] or [15:8] of Source Depth
9.0-9.7	255:0	Stencil
		Format: MDPR_STENCIL Slots [7:0] or [15:8] of Stencil

OS SZ SIMD8 Render Target Data Payload

MDP_RTW_SZ8 - OS SZ SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Red Format: MDP_DW_SIMD8 Slots [7:0] Red
1.0-1.7	255:0	Green Format: MDP_DW_SIMD8 Slots [7:0] Green
2.0-2.7	255:0	Blue Format: MDP_DW_SIMD8 Slots [7:0] Blue
3.0-3.7	255:0	Alpha Format: MDP_DW_SIMD8 Slots [7:0] Alpha
4.0-4.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] Source Depth
5.0-5.7	255:0	Stencil Format: MDPR_STENCIL Slots [7:0] Stencil

Oword 1 Dual Block Data Payload

MDP_OWD1 - Oword 1 Dual Block Data Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.3	127:0	Oword Slot0
		<table border="1"> <tr> <td>Format:</td> <td>U128</td> </tr> </table> Specifies the Slot 0 data
Format:	U128	
0.4-0.7	127:0	Oword Slot1
		<table border="1"> <tr> <td>Format:</td> <td>U128</td> </tr> </table> Specifies the Slot 1 data
Format:	U128	

Oword 2 Block Data Payload

MDP_OW2 - Oword 2 Block Data Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.3	127:0	Oword0
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U128</td> </tr> </table> <p>Specifies the Oword data for block element 0</p>
Format:	U128	
0.4-0.7	127:0	Oword1
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U128</td> </tr> </table> <p>Specifies the Oword data for block element 1</p>
Format:	U128	

Oword 4 Block Data Payload

MDP_OW4 - Oword 4 Block Data Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Data[1:0]
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDCR_OW</td> </tr> </table> <p>Specifies the Oword data for block elements [1:0]</p>
Format:	MDCR_OW	
1.0-1.7	255:0	Data[3:2]
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDCR_OW</td> </tr> </table> <p>Specifies the Oword data for block elements [3:2]</p>
Format:	MDCR_OW	

Oword A64 SIMD4x2 Atomic CMPWR16B Message Data Payload

MDP_A64_AOP4X2_OW2 - Oword A64 SIMD4x2 Atomic CMPWR16B Message Data Payload				
Source:	BSpec			
Size (in bits):	512			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.3	127:0	Src0 Slot0 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U128</td> </tr> </table> Specifies the Slot 0 Source 0 data	Format:	U128
Format:	U128			
0.4-0.7	127:0	Src0 Slot1 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U128</td> </tr> </table> Specifies the Slot 1 Source 0 data	Format:	U128
Format:	U128			
1.0-1.3	127:0	Src1 Slot0 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U128</td> </tr> </table> Specifies the Slot 0 Source 1 data	Format:	U128
Format:	U128			
1.4-1.7	127:0	Src1 Slot1 <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U128</td> </tr> </table> Specifies the Slot 1 Source 1 data	Format:	U128
Format:	U128			

Oword A64 SIMD4x2 Atomic Operation Return Data Message Data Payload

MDP_A64_AOP4X2_OW1 - Oword A64 SIMD4x2 Atomic Operation Return Data Message Data Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.3	127:0	Oword0 <table border="1"> <tr> <td>Format:</td> <td>U128</td> </tr> </table> Specifies the Slot 0 Return data	Format:	U128
Format:	U128			
0.4-0.7	127:0	Oword1 <table border="1"> <tr> <td>Format:</td> <td>U128</td> </tr> </table> Specifies the Slot1 Return data	Format:	U128
Format:	U128			



MDP_A64_AOP8_OW2 - Oword A64 SIMD8 Atomic Operation CMPWR16B Message Data Payload		
7.0-7.7	255:0	Slot[7:6] Src1
		<table border="1"> <tr> <td>Format:</td> <td>MDCR_OW</td> </tr> </table> <p>Specifies the Slot [7:6] Source 1 data</p>
Format:	MDCR_OW	

Oword Data Blocks Message Descriptor Control Field

MDC_DB_OW - Oword Data Blocks Message Descriptor Control Field																											
Source:	BSpec																										
Size (in bits):	3																										
Default Value:	0x00000000																										
DWord	Bit	Description																									
0	2:0	<p>Data Blocks</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> <tr> <td colspan="2">Specifies the number of Oword blocks to be read or written</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">OW1L</td> <td>1 Oword, read into or written from the low 128 bits of the destination register</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">OW1U</td> <td>1 Oword, read into or written from the high 128 bits of the destination register</td> </tr> <tr> <td style="text-align: center;">02h</td> <td style="text-align: center;">OW2</td> <td>2 Owords</td> </tr> <tr> <td style="text-align: center;">03h</td> <td style="text-align: center;">OW4</td> <td>4 Owords</td> </tr> <tr> <td style="text-align: center;">04h</td> <td style="text-align: center;">OW8</td> <td>8 Owords</td> </tr> <tr> <td style="text-align: center;">Others</td> <td style="text-align: center;">Reserved</td> <td>Ignored</td> </tr> </table>	Format:	Enumeration	Specifies the number of Oword blocks to be read or written		Value	Name	Description	00h	OW1L	1 Oword, read into or written from the low 128 bits of the destination register	01h	OW1U	1 Oword, read into or written from the high 128 bits of the destination register	02h	OW2	2 Owords	03h	OW4	4 Owords	04h	OW8	8 Owords	Others	Reserved	Ignored
Format:	Enumeration																										
Specifies the number of Oword blocks to be read or written																											
Value	Name	Description																									
00h	OW1L	1 Oword, read into or written from the low 128 bits of the destination register																									
01h	OW1U	1 Oword, read into or written from the high 128 bits of the destination register																									
02h	OW2	2 Owords																									
03h	OW4	4 Owords																									
04h	OW8	8 Owords																									
Others	Reserved	Ignored																									

Oword Data Payload Register

MDCR_OW - Oword Data Payload Register		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.3	127:0	Oword0
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U128</td> </tr> </table> <p>Specifies the slot 0 data in this payload register</p>
Format:	U128	
0.4-0.7	127:0	Oword1
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U128</td> </tr> </table> <p>Specifies the slot 1 data in this payload register</p>
Format:	U128	

Oword Dual Data Blocks Message Descriptor Control Field

MDC_DB_OWD - Oword Dual Data Blocks Message Descriptor Control Field																				
Source:	BSpec																			
Size (in bits):	2																			
Default Value:	0x00000000																			
DWord	Bit	Description																		
0	1:0	<p>OW Dual Data Blocks</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td colspan="2">Enumeration</td> </tr> <tr> <td colspan="3">Specifies the number of Oword Blocks to be read or written</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td>00h</td> <td>OWD1</td> <td>1 Hword register, 2 Owords</td> </tr> <tr> <td>02h</td> <td>OWD4</td> <td>4 Hword registers, 8 Owords</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </table>	Format:	Enumeration		Specifies the number of Oword Blocks to be read or written			Value	Name	Description	00h	OWD1	1 Hword register, 2 Owords	02h	OWD4	4 Hword registers, 8 Owords	Others	Reserved	Ignored
Format:	Enumeration																			
Specifies the number of Oword Blocks to be read or written																				
Value	Name	Description																		
00h	OWD1	1 Hword register, 2 Owords																		
02h	OWD4	4 Hword registers, 8 Owords																		
Others	Reserved	Ignored																		

PALETTE_ENTRY

PALETTE_ENTRY				
Source:	RenderCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:24	<p>Alpha</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Alpha channel value for this entry in the texture color palette.</p>	Format:	U8
	Format:	U8		
	23:16	<p>Red</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Red channel value for this entry in the texture color palette.</p>	Format:	U8
	Format:	U8		
15:8	<p>Green</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Green channel value for this entry in the texture color palette.</p>	Format:	U8	
Format:	U8			
7:0	<p>Blue</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Blue channel value for this entry in the texture color palette.</p>	Format:	U8	
Format:	U8			

Performance Counter Report Format 101b		
22	31:0	A-Cntr 18 (low dword)
23	31:0	A-Cntr 19 (low dword)
24	31:0	A-Cntr 20 (low dword)
25	31:0	A-Cntr 21 (low dword)
26	31:0	A-Cntr 22 (low dword)
27	31:0	A-Cntr 23 (low dword)
28	31:0	A-Cntr 24 (low dword)
29	31:0	A-Cntr 25 (low dword)
30	31:0	A-Cntr 26 (low dword)
31	31:0	A-Cntr 27 (low dword)
32	31:0	A-Cntr 28 (low dword)
33	31:0	A-Cntr 29 (low dword)
34	31:0	A-Cntr 30 (low dword)
35	31:0	A-Cntr 31 (low dword)
36	31:0	A-Cntr 32 (low dword)
37	31:0	A-Cntr 33 (low dword)
38	31:0	A-Cntr 34 (low dword)
39	31:0	A-Cntr 35 (low dword)
40	31:24	High byte of A3
	23:16	High byte of A2
	15:8	High byte of A1
	7:0	High byte of A0
41	31:24	High byte of A7
	23:16	High byte of A6
	15:8	High byte of A5
	7:0	High byte of A4
42	31:24	High byte of A11
	23:16	High byte of A10
	15:8	High byte of A9
	7:0	High byte of A8
43	31:24	High byte of A15
	23:16	High byte of A14
	15:8	High byte of A13
	7:0	High byte of A12
44	31:24	High byte of A19
	23:16	High byte of A18

Performance Counter Report Format 101b		
	15:8	High byte of A17
	7:0	High byte of A16
45	31:24	High byte of A23
	23:16	High byte of A22
	15:8	High byte of A21
	7:0	High byte of A20
46	31:24	High byte of A27
	23:16	High byte of A26
	15:8	High byte of A25
	7:0	High byte of A24
47	31:24	High byte of A31
	23:16	High byte of A30
	15:8	High byte of A29
	7:0	High byte of A28
48	31:0	B-Cntr 0
49	31:0	B-Cntr 1
50	31:0	B-Cntr 2
51	31:0	B-Cntr 3
52	31:0	B-Cntr 4
53	31:0	B-Cntr 5
54	31:0	B-Cntr 6
55	31:0	B-Cntr 7
56	31:0	C-Cntr 0
57	31:0	C-Cntr 1
58	31:0	C-Cntr 2
59	31:0	C-Cntr 3
60	31:0	C-Cntr 4
61	31:0	C-Cntr 5
62	31:0	C-Cntr 6
63	31:0	C-Cntr 7

Per Thread Scratch Space Message Header Control

MHC_PTSS - Per Thread Scratch Space Message Header Control							
Source:	BSpec						
Size (in bits):	32						
Default Value:	0x00000000						
DWord	Bit	Description					
0	31:4	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore			
	Format:	Ignore					
3:0	<p>Per Thread Scratch Space</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Specifies the amount of scratch space allowed to be used by this thread for messages in which the Binding Table Index is Stateless model, otherwise this field is ignored. The data port will use this to bounds check scratch space messages. Value range = [0,11] represents [1KB, 2MB] in powers of two.</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">Writes out of bounds will be ignored. Reads out of bounds will return 0.</td> </tr> </table>	Format:	U4	Programming Notes		Writes out of bounds will be ignored. Reads out of bounds will return 0.	
Format:	U4						
Programming Notes							
Writes out of bounds will be ignored. Reads out of bounds will return 0.							

Pixel Masked Media Block Message Header

MH_MBPM - Pixel Masked Media Block Message Header		
Source:	DataPort 1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	X Offset
		Format: S31
		X offset (in bytes) of the upper left corner of the block into the surface.
		Programming Notes
When Message Mode is set to PIXEL_MASK, this field must be a multiple of 32.		
1	31:0	Y Offset
		Format: S31
		Y offset (in rows) of the upper left corner of the block into the surface.
		Programming Notes
When Message Mode is set to PIXEL_MASK, this field must be a multiple of 4.		
2	31:0	Media Block Message Control
		Format: MHC_MBPM_CONTROL
Specifies the message subtype is Pixel Masked.		
3	31:0	Pixel Mask
		Format: U32
		Specifies the Pixel Mask for writes when Message Mode field is PIXEL_MASK.
		Programming Notes
The Pixel Mask applies to the 2x2 square tiles (UL, UR, LL, LR), which themselves tiled (UL, UR, LL, LR) and then repeated on the right for the remaining 16-bits to cover a 4 row 8 column area.		
4	31:0	FFTID
		Format: MHC_FFTID
Fixed Function Thread ID		
5-7	95:0	Reserved
		Format: Ignore
Ignored		

Pixel Masked Media Block Message Header Control

MHC_MBPM_CONTROL - Pixel Masked Media Block Message Header Control										
Source:	BSpec									
Size (in bits):	32									
Default Value:	0x00000000									
DWord	Bit	Description								
0	31:30	Message Mode								
		Format: Enumeration								
		Specifies the Media Block Write Message subtype is Pixel Masked.								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>01h</td> <td>PIXEL_MASK</td> <td>Use the Pixel Mask in the Message Header. The Block Height and Block Width are ignored and behave as if they are set to 4 rows and 32 bytes, respectively.</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Reserved.</td> </tr> </tbody> </table>	Value	Name	Description	01h	PIXEL_MASK	Use the Pixel Mask in the Message Header. The Block Height and Block Width are ignored and behave as if they are set to 4 rows and 32 bytes, respectively.	Others	Reserved
	Value	Name	Description							
	01h	PIXEL_MASK	Use the Pixel Mask in the Message Header. The Block Height and Block Width are ignored and behave as if they are set to 4 rows and 32 bytes, respectively.							
	Others	Reserved	Reserved.							
29	Reserved									
	Format: Ignore Ignored									
28:24	Sub-Register Offset									
	Format: U5 This field is ignored (reserved) for a media block write message.									
23:22	Reserved									
	Format: Ignore Ignored									
21:16	Block Height									
	Format: U6 This field is ignored (reserved) for a Pixel Masked media block write message.									
15:10	Reserved									
	Format: Ignore Ignored									
9:8	Register Pitch Control									
	Format: U2 This field is ignored (reserved) for a media block write message.									

MHC_MBPM_CONTROL - Pixel Masked Media Block Message Header Control

	7:6	Reserved
	Format:	Ignore
		Ignored
	5:0	Block Width
	Format:	U6
		This field is ignored (reserved) for a Pixel Masked media block write message.

Pixel Sample Mask Message Header Control

MHC_PSM - Pixel Sample Mask Message Header Control					
Source:		BSpec			
Size (in bits):		32			
Default Value:		0x0000FFFF			
DWord	Bit	Description			
0	31:16	Reserved			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore	
	Format:	Ignore			
	15:0	Pixel Sample Mask			
<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>0FFFFh Default</td> </tr> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>SIMD16 and SIMD8 messages. All 16 bits are used for SIMD16. For untyped SIMD8 messages, the low 8 bits of field are used. If the header is not delivered, this field defaults to all ones. This field is ignored for SIMD4x2 messages.</p>		Default Value:	0FFFFh Default	Format:	U16
Default Value:		0FFFFh Default			
Format:	U16				

Pixel Sample Mask Render Target Message Header Control

MHC_RT_PSM - Pixel Sample Mask Render Target Message Header Control								
Source:	BSpec							
Size (in bits):	32							
Default Value:	0x00000000							
DWord	Bit	Description						
0	31:16	Dispatched Pixel/Sample Enables <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>One bit per pixel (or sample within pixel) indicating which pixels/samples were originally enabled when the thread was dispatched. The Dispatched Pixel/Sample Enables must be unmodified from the ones sent when the pixel shader thread was initiated. If the Dispatched Pixel/Sample Enables are modified, behavior is undefined.</p> <table border="1"> <tr> <th colspan="2">Programming Notes</th> </tr> <tr> <td colspan="2">When operating in PER_SAMPLE mode these bits correspond to samples, not pixels. Each subspan slot (4 bits) corresponds to a specific sample location for the subspan. Note that in NUMSAMPLES_1 mode, a pixel and sample are synonymous. When operating in PER_PIXEL mode, this field is ignored, and instead the SampleEnableMask (obtained via bypass) are used to clear the Depth Scoreboard.</td> </tr> </table>	Format:	U16	Programming Notes		When operating in PER_SAMPLE mode these bits correspond to samples, not pixels. Each subspan slot (4 bits) corresponds to a specific sample location for the subspan. Note that in NUMSAMPLES_1 mode, a pixel and sample are synonymous. When operating in PER_PIXEL mode, this field is ignored, and instead the SampleEnableMask (obtained via bypass) are used to clear the Depth Scoreboard.	
		Format:	U16					
Programming Notes								
When operating in PER_SAMPLE mode these bits correspond to samples, not pixels. Each subspan slot (4 bits) corresponds to a specific sample location for the subspan. Note that in NUMSAMPLES_1 mode, a pixel and sample are synonymous. When operating in PER_PIXEL mode, this field is ignored, and instead the SampleEnableMask (obtained via bypass) are used to clear the Depth Scoreboard.								
	15:0	Pixel/Sample Enables <table border="1"> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>Specifies which pixels/samples are still lit based on kill instruction activity in the pixel shader. This mask is AND'd with the Dispatched Pixel/Sample Enables mask, and that is used to control actual accesses to the color buffer. Pixels/samples will be dropped on masked writes, and the GRF is not modified for masked reads.</p> <table border="1"> <tr> <th colspan="2">Programming Notes</th> </tr> <tr> <td colspan="2">When operating in PER_SAMPLE mode these bits correspond to samples, not pixels, as the PS is run per-sample. Each subspan slot (4 bits) corresponds to a specific sample location for the subspan. When operating in PER_PIXEL mode, these bits still correspond to pixels, as the PS is run per-pixel. Each pixel's mask bit is replicated according to Number of Multisamples and combined with other masks to control writes to the multisample locations.</td> </tr> </table>	Format:	U16	Programming Notes		When operating in PER_SAMPLE mode these bits correspond to samples, not pixels, as the PS is run per-sample. Each subspan slot (4 bits) corresponds to a specific sample location for the subspan. When operating in PER_PIXEL mode, these bits still correspond to pixels, as the PS is run per-pixel. Each pixel's mask bit is replicated according to Number of Multisamples and combined with other masks to control writes to the multisample locations.	
		Format:	U16					
Programming Notes								
When operating in PER_SAMPLE mode these bits correspond to samples, not pixels, as the PS is run per-sample. Each subspan slot (4 bits) corresponds to a specific sample location for the subspan. When operating in PER_PIXEL mode, these bits still correspond to pixels, as the PS is run per-pixel. Each pixel's mask bit is replicated according to Number of Multisamples and combined with other masks to control writes to the multisample locations.								

Power Clock State Format

Power Clock State Format			
Source:	RenderCS		
Size (in bits):	31		
Default Value:	0x00000088		
Known Uses <ul style="list-style-type: none"> • R_PWR_CLK_STATE - Render Power Clock State Register • PM_PWR_CLK_STATE - PM Power Clock State Request (Intended, in GT/GTI space, not yet in use) • PM_PWR_CLK_STATE (Intended, in GT/GTI space, not yet in use) 			
DWord	Bit	Description	
0	30:20	Reserved	
		Access:	RO
		Format:	MBZ
	19	Reserved	
		Access:	RO
		Format:	MBZ
	18	Enable Slice Count Request	
		Access:	R/W
		Enable Slice Count Request. This field is for Broadwell and is not used on Cherryview.	
		Value	Name Description
0h		Disable	Use async PMunit slice count request.
1h	Enable	Use SliceCount from this register.	
17:15	Slice Count Request		
	Access:	R/W	
	Slice Count Request. This field is for Broadwell and is not used on Cherryview. This field is for Skylake and is not used on Broxton.		
	This is further limited to the number of slices in a given SKU		
	Value	Name Description	
	001b		1 slice.
	010b		2 slices.
	011b		3 slices.
	100b		4 slices.
	101b		5 slices. Hardware will revert to 4 slices
110b		6 slices.	
8	Reserved		
	Access:	RO	

Power Clock State Format																	
7:4	EUmax																
	Access:	R/W															
Maximum number of EUs to power (per subslice if multiple subslices enabled). To specify an exact number of subslices, set EUmax equal to EUmin.																	
<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>0010b</td> <td></td> <td>2 EUs</td> </tr> <tr> <td>0100b</td> <td></td> <td>4 EUs</td> </tr> <tr> <td>0110b</td> <td></td> <td>6 EUs</td> </tr> <tr> <td>1000b</td> <td style="text-align: center;">[Default]</td> <td>8 EUs</td> </tr> </tbody> </table>			Value	Name	Description	0010b		2 EUs	0100b		4 EUs	0110b		6 EUs	1000b	[Default]	8 EUs
Value	Name	Description															
0010b		2 EUs															
0100b		4 EUs															
0110b		6 EUs															
1000b	[Default]	8 EUs															
Programming Notes																	
EUmin and EUmax need to be even and odd numbers are illegal; hardware will clip odd EU counts to an even value.																	
3:0	EUmin																
	Access:	R/W															
Minimum number of EUs to power (per subslice if multiple subslices enabled). To specify an exact number of subslices, set EUmax equal to EUmin.																	
<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>0010b</td> <td></td> <td>2 EUs</td> </tr> <tr> <td>0100b</td> <td></td> <td>4 EUs</td> </tr> <tr> <td>0110b</td> <td></td> <td>6 EUs</td> </tr> <tr> <td>1000b</td> <td style="text-align: center;">[Default]</td> <td>8 EUs</td> </tr> </tbody> </table>			Value	Name	Description	0010b		2 EUs	0100b		4 EUs	0110b		6 EUs	1000b	[Default]	8 EUs
Value	Name	Description															
0010b		2 EUs															
0100b		4 EUs															
0110b		6 EUs															
1000b	[Default]	8 EUs															
Programming Notes																	
EUmin and EUmax need to be even and odd numbers are illegal; hardware will clip odd EU counts to an even value.																	

PPHWSP_LAYOUT - PPHWSP_LAYOUT		
32..33	63:0	Reserved
1020-34	31:0	Reserved

Qword A64 SIMD4x2 Atomic CMPWR Message Data Payload

MDP_A64_AOP4X2_QW2 - Qword A64 SIMD4x2 Atomic CMPWR Message Data Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.1	63:0	Src0 Slot0
		Format: U64 Specifies the Slot 0 Source 0 data
0.2-0.3	63:0	Reserved
		Format: Ignore Ignored
0.4-0.5	63:0	Src0 Slot1
		Format: U64 Specifies the Slot 1 Source 0 data
0.6-0.7	63:0	Reserved
		Format: Ignore Ignored
1.0-1.1	63:0	Src1 Slot0
		Format: U64 Specifies the Slot 0 Source 1 data
1.2-1.3	63:0	Reserved
		Format: Ignore Ignored
1.4-1.5	63:0	Src1 Slot1
		Format: U64 Specifies the Slot 1 Source 1 data
1.6-1.7	63:0	Reserved
		Format: Ignore Ignored

Qword Data Payload Register

MDCR_QW - Qword Data Payload Register		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.1	63:0	Qword0
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U64</td> </tr> </table> <p>Specifies the slot 0 data in this payload register</p>
Format:	U64	
0.2-0.3	63:0	Qword1
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U64</td> </tr> </table> <p>Specifies the slot 1 data in this payload register</p>
Format:	U64	
0.4-0.5	63:0	Qword2
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U64</td> </tr> </table> <p>Specifies the slot 2 data in this payload register</p>
Format:	U64	
0.6-0.7	63:0	Qword3
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U64</td> </tr> </table> <p>Specifies the slot 3 data in this payload register</p>
Format:	U64	

Qword SIMD4x2 Atomic CMPWR8B Message Data Payload

MDP_AOP4X2_QW2 - Qword SIMD4x2 Atomic CMPWR8B Message Data Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0-1	63:0	Src0 Slot0
		Format: U64 Specifies the Slot 0 Source 0 data
2-3	63:0	Src1 Slot0
		Format: U64 Specifies the Slot 0 Source 1 data
4-5	63:0	Src0 Slot1
		Format: U64 Specifies the Slot 1 Source 0 data
6-7	63:0	Src1 Slot1
		Format: U64 Specifies the Slot 1 Source 1 data

Qword SIMD4x2 Atomic Operation Message Data Payload

MDP_AOP4X2_QW1 - Qword SIMD4x2 Atomic Operation Message Data Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0-1	63:0	<p>Qword0</p> <table border="1"> <tr> <td>Format:</td> <td>U64 S63</td> </tr> </table> <p>Specifies the Slot 0 Source or Return data</p>	Format:	U64 S63
Format:	U64 S63			
2-3	63:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
4-5	63:0	<p>Qword1</p> <table border="1"> <tr> <td>Format:</td> <td>U64 S63</td> </tr> </table> <p>Specifies the Slot 1 Source or Return data</p>	Format:	U64 S63
Format:	U64 S63			
6-7	63:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

Qword SIMD8 Atomic Operation Return Data Message Data Payload

MDP_AOP8_QW1 - Qword SIMD8 Atomic Operation Return Data Message Data Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Slot[7:0] Qword[31:0]
		<table border="1"> <tr> <td>Format:</td> <td>MDCR_DW</td> </tr> </table> <p>Specifies the lower 32-bits of Slot [7:0] Return data</p>
Format:	MDCR_DW	
1.0-1.7	255:0	Slot[7:0] Qword[63:32]
		<table border="1"> <tr> <td>Format:</td> <td>MDCR_DW</td> </tr> </table> <p>Specifies the upper 32-bits of Slot [7:0] Return data</p>
Format:	MDCR_DW	

Qword SIMD8 Data Payload

MDP_QW_SIMD8 - Qword SIMD8 Data Payload				
Source:	BSpec			
Size (in bits):	512			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	Data[3:0] <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDCR_QW</td> </tr> </table> Specifies the Slot [3:0] data	Format:	MDCR_QW
Format:	MDCR_QW			
1.0-1.7	255:0	Data[7:4] <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MDCR_QW</td> </tr> </table> Specifies the Slot [7:4] data	Format:	MDCR_QW
Format:	MDCR_QW			

MDP_AOP16_QW2 - Qword SIMD16 Atomic Operation CMPWR8B Message Data Payload				
6.0-6.7	255:0	Slot[7:0] Src1[63:32] <table border="1"> <tr> <td>Format:</td> <td>MDCR_DW</td> </tr> </table> Specifies the upper 32-bits of Source 1 data for Slot [7:0]	Format:	MDCR_DW
Format:	MDCR_DW			
7.0-7.7	255:0	Slot[15:8] Src1[63:32] <table border="1"> <tr> <td>Format:</td> <td>MDCR_DW</td> </tr> </table> Specifies the upper 32-bits Source 1 data for Slot [15:8]	Format:	MDCR_DW
Format:	MDCR_DW			

Qword SIMD16 Data Payload

MDP_QW_SIMD16 - Qword SIMD16 Data Payload		
Source:	BSpec	
Size (in bits):	1024	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Data[3:0] Format: MDCR_QW Specifies the Slot [3:0] data
1.0-1.7	255:0	Data[7:4] Format: MDCR_QW Specifies the Slot [7:4] data
2.0-2.7	255:0	qw11_qw8 Format: MDCR_QW Specifies the Slot [11:8] data
3.0-3.7	255:0	qw15_qw12 Format: MDCR_QW Specifies the Slot [15:12] data

Read-Only Data Port Message Types

MT_DP_RO - Read-Only Data Port Message Types				
Source:	Read-Only DataPort			
Size (in bits):	5			
Default Value:	0x00000000			
Lists all the Message Types in a Read-Only Data Port Message Descriptor [18:14]. Read operations from the Constant Cache and Sampler Cache are encoded in the Read-Only Data Port. Many of the operations are also implemented in Data Port 0, and those operations use the same Message Header.				
DWord	Bit	Description		
0	4	Reserved		
		Format: MBZ Ignored		
	3:0	Message Type		
		Format: Enumeration		
		Specifies type of message		
		Value	Name	Description
		00h	MT_CC_OWb [Default]	Oword Block Read Constant Cache message
		01h	MT_CC_OWUB	Unaligned Oword Block Read Constant Cache message
		02h	MT_CC_OWDB	Oword Dual Block Read Constant Cache message
		03h	MT_CC_DWS	Dword Scattered Read Constant Cache message
04h	MT_SC_OWUB	Unaligned Oword Block Read Sampler Cache message		
05h	MT_SC_MB	Media Block Read Sampler Cache message		
06h	MT_RSI	Read Surface Info message		
Others	Reserved	Ignored		

Read Surface Info 32-Bit Address Payload

MAP32B_RSI - Read Surface Info 32-Bit Address Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0	31:0	U
		Format: U32 Specifies the U channel address offset.
0.1	31:0	V
		Format: U32 Specifies the V channel address offset.
0.2	31:0	R
		Format: U32 Specifies the R channel address offset.
0.3	31:0	LOD
		Format: MACD_LOD Specifies the LOD.
0.4-0.7	127:0	Reserved
		Format: Ignore Ignored

Read Surface Info Data Payload

MDP_RSI - Read Surface Info Data Payload						
Source:	BSpec					
Size (in bits):	512					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000					
DWord	Bit	Description				
0.0-0.5	191:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore		
Format:	Ignore					
0.6-0.7	63:0	<p>Instruction Base Address</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>GraphicsAddress[63:0]</td> </tr> </table> <p>Instruction Base Address from STATE_BASE_ADDRESS, extended to 64-bit format.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> </table> <p>The 48-bit address is returned in a 64-bit address in canonical form.</p>	Format:	GraphicsAddress[63:0]	Programming Notes	
Format:	GraphicsAddress[63:0]					
Programming Notes						
1.0	31:0	<p>Width</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Surface Width generally computed from RENDER_SURFACE_STATE Width (stored as width minus 1). The value is 0 for NULL surface, and in all other cases (Width+1) » LOD. Surface Width from RENDER_SURFACE_STATE (U14), zero extended to 32 bits.</p>	Format:	U32		
Format:	U32					
1.1	31:0	<p>Height</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Surface Height, generally computed from RENDER_SURFACE_STATE Height (stored as height minus 1). The value for a 1D array is RENDER_SURFACE_STATE's (Depth + 1). The value for 1D non-array, BUFFER, and NULL surface is 0. In all other case, the value is (Height + 1) » LOD.</p>	Format:	U32		
Format:	U32					
1.2	31:0	<p>Depth</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Surface Depth, generally computed from RENDER_SURFACE_STATE Depth (which is stored depth minus 1). If 2D Array or Cube Array surface, value is the (Depth+1). If 3D surface, value is (Depth+1) » LOD. In all other case, the value is 0.</p>	Format:	U32		
Format:	U32					
1.3	31:0	<p>MIP Count</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>MIP Count from RENDER_SURFACE_STATE, range [0, 14], zero extended to 32 bits.</p>	Format:	U32		
Format:	U32					

MDP_RSI - Read Surface Info Data Payload																													
1.4	31:0	Surface Type																											
		Format: U32																											
		Surface Type from RENDER_SURFACE_STATE, zero extended to 32 bits																											
		<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>SURFTYPE_1D</td> <td>1-dimensional map or array of maps</td> </tr> <tr> <td>1h</td> <td>SURFTYPE_2D</td> <td>2-dimensional map or array of maps</td> </tr> <tr> <td>2h</td> <td>SURFTYPE_3D</td> <td>3-dimensional map (volumetric) of maps</td> </tr> <tr> <td>3h</td> <td>SURFTYPE_CUBE</td> <td>Cube map or array of cube maps</td> </tr> <tr> <td>4h</td> <td>SURFTYPE_BUFFER</td> <td>Element in a buffer</td> </tr> <tr> <td>5h</td> <td>SURFTYPE_STRBUF</td> <td>Structured buffer surface</td> </tr> <tr> <td>7h</td> <td>SURTYPE_NULL</td> <td>Null surface</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	0h	SURFTYPE_1D	1-dimensional map or array of maps	1h	SURFTYPE_2D	2-dimensional map or array of maps	2h	SURFTYPE_3D	3-dimensional map (volumetric) of maps	3h	SURFTYPE_CUBE	Cube map or array of cube maps	4h	SURFTYPE_BUFFER	Element in a buffer	5h	SURFTYPE_STRBUF	Structured buffer surface	7h	SURTYPE_NULL	Null surface	Others	Reserved	Reserved
		Value	Name	Description																									
		0h	SURFTYPE_1D	1-dimensional map or array of maps																									
		1h	SURFTYPE_2D	2-dimensional map or array of maps																									
		2h	SURFTYPE_3D	3-dimensional map (volumetric) of maps																									
		3h	SURFTYPE_CUBE	Cube map or array of cube maps																									
		4h	SURFTYPE_BUFFER	Element in a buffer																									
5h	SURFTYPE_STRBUF	Structured buffer surface																											
7h	SURTYPE_NULL	Null surface																											
Others	Reserved	Reserved																											
0h	SURFTYPE_1D	1-dimensional map or array of maps																											
1h	SURFTYPE_2D	2-dimensional map or array of maps																											
2h	SURFTYPE_3D	3-dimensional map (volumetric) of maps																											
3h	SURFTYPE_CUBE	Cube map or array of cube maps																											
4h	SURFTYPE_BUFFER	Element in a buffer																											
5h	SURFTYPE_STRBUF	Structured buffer surface																											
7h	SURTYPE_NULL	Null surface																											
Others	Reserved	Reserved																											
1.5	31:0	Surface Format																											
		Format: U32 Surface Format from RENDER_SURFACE_STATE (U9), zero extended to 32 bits.																											
1.6-1.7	63:0	Reserved																											
		Format: Ignore Ignored																											

RENDER_SURFACE_STATE			
28	<p>Surface Array</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field, if enabled, indicates that the surface is an array.</p> <p style="text-align: center;">Programming Notes</p> <p>If this field is <i>enabled</i>, the Surface Type must be SURFTYPE_1D, SURFTYPE_2D, or SURFTYPE_CUBE. If this field is <i>disabled</i> and Surface Type is SURFTYPE_1D, SURFTYPE_2D, or SURFTYPE_CUBE, the Depth field must be set to zero.</p>	Format:	Enable
Format:	Enable		
27	<p>ASTC_Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field, if enabled, indicates that the surface is one of ASTC compression formats.</p> <p style="text-align: center;">Programming Notes</p> <p>If this field is <i>enabled</i>, the definition of Surface Format encoding will follow a new convention defined by ASTC. If this field is <i>disabled</i>, the definition of Surface Format will follow the legacy convention defined in non-ASTC style.</p>	Format:	Enable
Format:	Enable		
26:18	<p>Surface Format</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>SURFACE_FORMAT</td> </tr> </table> <p style="text-align: center;">Programming Notes</p> <p>If ASTC_Enable is set to 0: YUV (YCRCB) surfaces used as render targets can only be rendered to using 3DPRIM_RECTLIST with even X coordinates on all of its vertices, and the pixel shader cannot kill pixels.</p> <p>If Number of Multisamples is set to a value other than MULTISAMPLECOUNT_1, this field cannot be set to the following formats:</p> <ul style="list-style-type: none"> • Any compressed texture format (BC*, DXT*, FXT*, ETC*, EAC*) • Any YCRCB* format <p>If ASTC_Enable is set to 1:</p> <ul style="list-style-type: none"> • ASTC_Profile can be freely set to 1 or 0 with full ASTC profile supported in SKL products. • ASTC_BlockDimention: Bit [25] will be set to 0 if ASTC_Profile is LDR-profile since only 2D blocks are supported in LDR profile. This is the case for SKL. In SKL, 2D HDR-profile can be supported where the ASTC_Profile is set at 1 (Full-Profile) and ASTC_BlockDimention is set as 0 (2D). <p>This field cannot ASTC format if the Surface Type is SURFTYPE_BUFFER or SURFTYPE_STRBUF This field cannot be ASTC format if the Surface Type is SURFTYPE_1D.</p> <p>This field cannot be a YUV (YCRCB*) or compressed (BC*, DXT*, FXT*, ETC*, EAC*) format if the Surface Type is SURFTYPE_BUFFER or SURFTYPE_STRBUF This field cannot be a compressed (BC*, DXT*, FXT*, ETC*, EAC*) format if the Surface Type is SURFTYPE_1D.</p>	Format:	SURFACE_FORMAT
Format:	SURFACE_FORMAT		

RENDER_SURFACE_STATE

17:16	Surface Vertical Alignment	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3" style="text-align: center; background-color: #e6f2ff;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="3"> <p>For Sampling Engine and Render Target Surfaces: This field specifies the vertical alignment requirement in elements for the surface. Refer to the "Memory Data Formats" chapter for details on how this field changes the layout of the surface in memory. An <i>element</i> is defined as a pixel in uncompressed surface formats, and as a compression block in compressed surface formats. For MSFMT_DEPTH_STENCIL type multisampled surfaces, an element is a sample.</p> <p>This field is used for 2D, CUBE, and 3D surface alignment when Tiled Resource Mode is TRMODE_NONE (Tiled Resource Mode is disabled). This field is ignored for 1D surfaces and also when Tiled Resource Mode is not TRMODE_NONE (e.g. Tiled Resource Mode is enabled). See the appropriate Alignment table in the "Surface Layout and Tiling" section under Common Surface Formats for the table of alignment values for Tiled Resources.</p> <p>For other surfaces: This field is ignored.</p> </td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Reserved</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">VALIGN 4</td> <td style="text-align: center;">Vertical alignment factor j = 4</td> </tr> <tr> <td style="text-align: center;">2h</td> <td style="text-align: center;">VALIGN 8</td> <td style="text-align: center;">Vertical alignment factor j = 8</td> </tr> <tr> <td style="text-align: center;">3h</td> <td style="text-align: center;">VALIGN 16</td> <td style="text-align: center;">Vertical alignment factor j = 16</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="1" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td> <p>This field is intended to be set to VALIGN_4 if the surface was rendered as a depth buffer, for a multisampled (4x) render target, or for a multisampled (8x) render target, since these surfaces support only alignment of 4. Use of VALIGN_4 for other surfaces is supported, but increases memory usage.</p> <p>This field is intended to be set to VALIGN_8 only if the surface was rendered as a stencil buffer, since stencil buffer surfaces support only alignment of 8. If set to VALIGN_8, Surface Format must be R8_UINT.</p> <p>For uncompressed surfaces, the units of "j" are rows of pixels on the physical surface. For compressed texture formats, the units of "j" are in compression blocks, thus each increment in "j" is equal to h pixels, where h is the height of the compression block in pixels.</p> </td> </tr> </tbody> </table>	Description			<p>For Sampling Engine and Render Target Surfaces: This field specifies the vertical alignment requirement in elements for the surface. Refer to the "Memory Data Formats" chapter for details on how this field changes the layout of the surface in memory. An <i>element</i> is defined as a pixel in uncompressed surface formats, and as a compression block in compressed surface formats. For MSFMT_DEPTH_STENCIL type multisampled surfaces, an element is a sample.</p> <p>This field is used for 2D, CUBE, and 3D surface alignment when Tiled Resource Mode is TRMODE_NONE (Tiled Resource Mode is disabled). This field is ignored for 1D surfaces and also when Tiled Resource Mode is not TRMODE_NONE (e.g. Tiled Resource Mode is enabled). See the appropriate Alignment table in the "Surface Layout and Tiling" section under Common Surface Formats for the table of alignment values for Tiled Resources.</p> <p>For other surfaces: This field is ignored.</p>			Value	Name	Description	0h	Reserved	Reserved	1h	VALIGN 4	Vertical alignment factor j = 4	2h	VALIGN 8	Vertical alignment factor j = 8	3h	VALIGN 16	Vertical alignment factor j = 16	Programming Notes	<p>This field is intended to be set to VALIGN_4 if the surface was rendered as a depth buffer, for a multisampled (4x) render target, or for a multisampled (8x) render target, since these surfaces support only alignment of 4. Use of VALIGN_4 for other surfaces is supported, but increases memory usage.</p> <p>This field is intended to be set to VALIGN_8 only if the surface was rendered as a stencil buffer, since stencil buffer surfaces support only alignment of 8. If set to VALIGN_8, Surface Format must be R8_UINT.</p> <p>For uncompressed surfaces, the units of "j" are rows of pixels on the physical surface. For compressed texture formats, the units of "j" are in compression blocks, thus each increment in "j" is equal to h pixels, where h is the height of the compression block in pixels.</p>
Description																									
<p>For Sampling Engine and Render Target Surfaces: This field specifies the vertical alignment requirement in elements for the surface. Refer to the "Memory Data Formats" chapter for details on how this field changes the layout of the surface in memory. An <i>element</i> is defined as a pixel in uncompressed surface formats, and as a compression block in compressed surface formats. For MSFMT_DEPTH_STENCIL type multisampled surfaces, an element is a sample.</p> <p>This field is used for 2D, CUBE, and 3D surface alignment when Tiled Resource Mode is TRMODE_NONE (Tiled Resource Mode is disabled). This field is ignored for 1D surfaces and also when Tiled Resource Mode is not TRMODE_NONE (e.g. Tiled Resource Mode is enabled). See the appropriate Alignment table in the "Surface Layout and Tiling" section under Common Surface Formats for the table of alignment values for Tiled Resources.</p> <p>For other surfaces: This field is ignored.</p>																									
Value	Name	Description																							
0h	Reserved	Reserved																							
1h	VALIGN 4	Vertical alignment factor j = 4																							
2h	VALIGN 8	Vertical alignment factor j = 8																							
3h	VALIGN 16	Vertical alignment factor j = 16																							
Programming Notes																									
<p>This field is intended to be set to VALIGN_4 if the surface was rendered as a depth buffer, for a multisampled (4x) render target, or for a multisampled (8x) render target, since these surfaces support only alignment of 4. Use of VALIGN_4 for other surfaces is supported, but increases memory usage.</p> <p>This field is intended to be set to VALIGN_8 only if the surface was rendered as a stencil buffer, since stencil buffer surfaces support only alignment of 8. If set to VALIGN_8, Surface Format must be R8_UINT.</p> <p>For uncompressed surfaces, the units of "j" are rows of pixels on the physical surface. For compressed texture formats, the units of "j" are in compression blocks, thus each increment in "j" is equal to h pixels, where h is the height of the compression block in pixels.</p>																									
15:14	Surface Horizontal Alignment	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="1" style="text-align: center; background-color: #e6f2ff;">Description</th> </tr> </thead> <tbody> <tr> <td> <p>For Sampling Engine and Render Target Surfaces: This field specifies the horizontal alignment requirement for the surface.</p> <p>This field is used for alignment when LOD >= Mip Tail Start LOD</p> <p>This field is ignored when Tiled Resource Mode is not TRMODE_NONE (i.e. Tiled Resources are enabled). See the "Surface Layout and Tiling" section under Common Surface Formats for the table of alignment values for Tile Resources.</p> <p>For other surfaces: This field is ignored.</p> </td> </tr> </tbody> </table>	Description	<p>For Sampling Engine and Render Target Surfaces: This field specifies the horizontal alignment requirement for the surface.</p> <p>This field is used for alignment when LOD >= Mip Tail Start LOD</p> <p>This field is ignored when Tiled Resource Mode is not TRMODE_NONE (i.e. Tiled Resources are enabled). See the "Surface Layout and Tiling" section under Common Surface Formats for the table of alignment values for Tile Resources.</p> <p>For other surfaces: This field is ignored.</p>																					
Description																									
<p>For Sampling Engine and Render Target Surfaces: This field specifies the horizontal alignment requirement for the surface.</p> <p>This field is used for alignment when LOD >= Mip Tail Start LOD</p> <p>This field is ignored when Tiled Resource Mode is not TRMODE_NONE (i.e. Tiled Resources are enabled). See the "Surface Layout and Tiling" section under Common Surface Formats for the table of alignment values for Tile Resources.</p> <p>For other surfaces: This field is ignored.</p>																									

RENDER_SURFACE_STATE			
	Value	Name	Description
	0h	Reserved	Reserved
	1h	HALIGN 4	Horizontal alignment factor j = 4
	2h	HALIGN 8	Horizontal alignment factor j = 8
	3h	HALIGN 16	Horizontal alignment factor j = 16
Programming Notes			
<p>This field is intended to be set to HALIGN_8 only if the surface was rendered as a depth buffer with Z16 format or a stencil buffer. In this case it must be set to HALIGN_8 since these surfaces support only alignment of 8. For Z32 formats it must be set to HALIGN_4. Use of HALIGN_8 for other surfaces is supported, but increases memory usage.</p> <p>For uncompressed surfaces, the units of "i" are pixels on the physical surface. For compressed texture formats, the units of "i" are in compression blocks, thus each increment in "i" is equal to w pixels, where w is the width of the compression block in pixels.</p> <p>When Auxiliary Surface Mode is set to AUX_CCS_D or AUX_CCS_E, HALIGN 16 must be used.</p>			
13:12	Tile Mode This field specifies the type of memory tiling (Linear, WMajor, XMajor, or YMajor) employed to tile this surface. See <i>Memory Interface Functions</i> for details on memory tiling and restrictions.		
	Value	Name	Description
	0h	LINEAR	Linear mode (no tiling)
	1h	WMAJOR	W major tiling
	2h	XMAJOR	X major tiling
	3h	YMAJOR	Y major tiling
Programming Notes			
<ul style="list-style-type: none"> Refer to <i>Memory Data Formats</i> for restrictions on <i>TileMode</i> direction for the various buffer types. (Of particular interest is the fact that YMAJOR tiling is not supported for display/overlay buffers). The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this field. Use of WMAJOR is valid only for sampling engine, Data Cache Data Port and render target surfaces and Surface Format must be R8_UINT. Vertical Line Stride must be zero. In addition to W tiling, this mode implies that the surface is stored as a stencil buffer. Refer to <i>Memory Data Formats</i> section for details on stencil buffer surface layout. Linear surfaces can be mapped to Main Memory (uncached) or System Memory (cacheable, snooped). Tiled (X/Y/W) surfaces can only be mapped to Main Memory. If Surface Type is SURFTYPE_BUFFER, this field must be TILEMODE_LINEAR If Number of Multisamples is not MULTISAMPLECOUNT_1, this field must be YMAJOR. 			
If Surface Type is SURFTYPE_STRBUF, this field must be TILEMODE_LINEAR.			
If Surface Type is SURFTYPE_1D this field must be TILEMODE_LINEAR, unless Sampler Legacy			

RENDER_SURFACE_STATE													
	<p>1D Map Layout Disable is set to 0, in which case TILEMODE_YMAJOR and TILEMODE_WMAJOR are also allowed. Tiled Resource Mode must be set to TRMODE_NONE for these cases.</p> <p>TILEMODE_XMAJOR is only allowed if Surface Type is SURFTYPE_2D.</p> <p>If Surface Format is ASTC*, this field must be TILEMODE_YMAJOR.</p>												
11	<p>Vertical Line Stride</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>U1 In lines to skip between logically adjacent lines</td> </tr> </table> <p>For 2D Non-Array Surfaces accessed via the Sampling Engine or Data Cache Data Port: Specifies number of lines (0 or 1) to skip between logically adjacent lines - provides support of interleaved (field) surfaces as textures.</p> <p>For Other Surfaces: Vertical Line Stride must be zero.</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">This bit must not be set if the surface format is a compressed type (BCn*, FXT1, ETC*, EAC*).</td> </tr> <tr> <td colspan="2">This bit must not be set if the surface format is compressed type ASTC*.</td> </tr> <tr> <td colspan="2">This bit must not be set if the Auxiliary Surface Mode is not AUX_NONE.</td> </tr> <tr> <td colspan="2">If this bit is set on a sampling engine surface, the mip mode filter must be set to MIPFILTER_NONE and the min and mag mode filter cannot be set to MAPFILTER_FLEXIBLE.</td> </tr> </table>	Format:	U1 In lines to skip between logically adjacent lines	Programming Notes		This bit must not be set if the surface format is a compressed type (BCn*, FXT1, ETC*, EAC*).		This bit must not be set if the surface format is compressed type ASTC*.		This bit must not be set if the Auxiliary Surface Mode is not AUX_NONE.		If this bit is set on a sampling engine surface, the mip mode filter must be set to MIPFILTER_NONE and the min and mag mode filter cannot be set to MAPFILTER_FLEXIBLE.	
Format:	U1 In lines to skip between logically adjacent lines												
Programming Notes													
This bit must not be set if the surface format is a compressed type (BCn*, FXT1, ETC*, EAC*).													
This bit must not be set if the surface format is compressed type ASTC*.													
This bit must not be set if the Auxiliary Surface Mode is not AUX_NONE.													
If this bit is set on a sampling engine surface, the mip mode filter must be set to MIPFILTER_NONE and the min and mag mode filter cannot be set to MAPFILTER_FLEXIBLE.													
10	<p>Vertical Line Stride Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>U1 In lines of initial offset (when Vertical Line Stride == 1)</td> </tr> </table> <p>For 2D Non-Array Surfaces accessed via the Sampling Engine or Data Cache Data Port: Specifies the offset of the initial line from the beginning of the buffer. Ignored when Vertical Line Stride is 0.</p> <p>For Other Surfaces: Vertical Line Stride Offset must be zero.</p>	Format:	U1 In lines of initial offset (when Vertical Line Stride == 1)										
Format:	U1 In lines of initial offset (when Vertical Line Stride == 1)												
9	<p>Sampler L2 out of order mode Disable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 55%;">Format:</td> <td>Disable</td> </tr> </table> <p>If disabled this will forced formats which would have bypassed the L2 and been filled into the L1 out of order to be cached in the L2 and send in order to the L1. In general that is any format which is expanded 1:2 in L1 or not expanded at all. This would include all lossless compressed cases For all other formats this will have no affect.</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">This bit must be set for the following surface types: BC2_UNORM BC3_UNORM BC5_UNORM BC5_SNORM BC7_UNORM</td> </tr> <tr> <td colspan="2">This bit must be set for surfaces which contain a HiZ auxilliary surface if other surfaces using AUX_CCS_E or AUX_CCS_D auxiliary surface state (lossless color compression) are being sampled at the same time.</td> </tr> </table>	Format:	Disable	Programming Notes		This bit must be set for the following surface types: BC2_UNORM BC3_UNORM BC5_UNORM BC5_SNORM BC7_UNORM		This bit must be set for surfaces which contain a HiZ auxilliary surface if other surfaces using AUX_CCS_E or AUX_CCS_D auxiliary surface state (lossless color compression) are being sampled at the same time.					
Format:	Disable												
Programming Notes													
This bit must be set for the following surface types: BC2_UNORM BC3_UNORM BC5_UNORM BC5_SNORM BC7_UNORM													
This bit must be set for surfaces which contain a HiZ auxilliary surface if other surfaces using AUX_CCS_E or AUX_CCS_D auxiliary surface state (lossless color compression) are being sampled at the same time.													
8	<p>Render Cache Read Write Mode</p> <p>For Surfaces accessed via the Data Port to Render Cache: This field specifies the way Render Cache treats a write request. If unset, Render Cache allocates a write-only cache line for a write miss. If set, Render Cache allocates a read-write cache line for a write miss.</p>												

RENDER_SURFACE_STATE																	
		<p>For Surfaces accessed via the Sampling Engine or Data Port to Texture Cache or Data Cache: This field is reserved : MBZ</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Write-Only Cache</td> <td>Allocating write-only cache for a write miss</td> </tr> <tr> <td>1h</td> <td>Read-Write Cache</td> <td>Allocating read-write cache for a write miss</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>This field is provided for performance optimization for Render Cache read/write accesses (from Gen4 EU's point of view).</p>	Value	Name	Description	0h	Write-Only Cache	Allocating write-only cache for a write miss	1h	Read-Write Cache	Allocating read-write cache for a write miss						
Value	Name	Description															
0h	Write-Only Cache	Allocating write-only cache for a write miss															
1h	Read-Write Cache	Allocating read-write cache for a write miss															
7:6	<p>Media Boundary Pixel Mode For 2D Non-Array Surfaces accessed via the Data Port Media Block Read Message or Data Port Transpose Read message: This field enables control of which rows are returned on vertical out-of-bounds reads using the Data Port Media Block Read Message or Data Port Transpose Read message. In the description below, frame mode refers to Vertical Line Stride = 0, field mode is Vertical Line Stride = 1 in which only the even or odd rows are addressable. The frame refers to the entire surface, while the field refers only to the even or odd rows within the surface.</p> <p>For Other Surfaces: Reserved : MBZ</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NORMAL_MODE</td> <td>The row returned on an out-of-bound access is the closest row in the frame or field. Rows from the opposite field are never returned.</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>2h</td> <td>PROGRESSIVE_FRAME</td> <td>The row returned on an out-of-bound access is the closest row in the frame, even if in field mode.</td> </tr> <tr> <td>3h</td> <td>INTERLACED_FRAME</td> <td>In field mode, the row returned on an out-of-bound access is the closest row in the field. In frame mode, even out-of-bound rows return the nearest even row while odd out-of-bound rows return the nearest odd row.</td> </tr> </tbody> </table>	Value	Name	Description	0h	NORMAL_MODE	The row returned on an out-of-bound access is the closest row in the frame or field. Rows from the opposite field are never returned.	1h	Reserved		2h	PROGRESSIVE_FRAME	The row returned on an out-of-bound access is the closest row in the frame, even if in field mode.	3h	INTERLACED_FRAME	In field mode, the row returned on an out-of-bound access is the closest row in the field. In frame mode, even out-of-bound rows return the nearest even row while odd out-of-bound rows return the nearest odd row.	
Value	Name	Description															
0h	NORMAL_MODE	The row returned on an out-of-bound access is the closest row in the frame or field. Rows from the opposite field are never returned.															
1h	Reserved																
2h	PROGRESSIVE_FRAME	The row returned on an out-of-bound access is the closest row in the frame, even if in field mode.															
3h	INTERLACED_FRAME	In field mode, the row returned on an out-of-bound access is the closest row in the field. In frame mode, even out-of-bound rows return the nearest even row while odd out-of-bound rows return the nearest odd row.															
5	<p>Cube Face Enable - Negative X</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Surface Type] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine: This field enable the individual face of a cube map. Enabling a face indicates that the face is present in the cube map, while disabling it indicates that that face is represented by the texture map's border color. Refer to Memory Data Formats for the correlation between faces and the cube map memory layout. Note that storage for disabled faces must be provided.</p> <p style="text-align: center;">Programming Notes</p> <p>When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</p>	Exists If:	[Surface Type] == 'SURFTYPE_CUBE'	Format:	Enable												
Exists If:	[Surface Type] == 'SURFTYPE_CUBE'																
Format:	Enable																

RENDER_SURFACE_STATE							
4	<p>Cube Face Enable - Positive X</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine: This field enable the individual face of a cube map. Enabling a face indicates that the face is present in the cube map, while disabling it indicates that that face is represented by the texture map's border color. Refer to Memory Data Formats for the correlation between faces and the cube map memory layout. Note that storage for disabled faces must be provided.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> </table>	Exists If:	[Surface Type] == 'SURFTYPE_CUBE'	Format:	Enable	Programming Notes	When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).
Exists If:	[Surface Type] == 'SURFTYPE_CUBE'						
Format:	Enable						
Programming Notes							
When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).							
3	<p>Cube Face Enable - Negative Y</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine: This field enable the individual face of a cube map. Enabling a face indicates that the face is present in the cube map, while disabling it indicates that that face is represented by the texture map's border color. Refer to Memory Data Formats for the correlation between faces and the cube map memory layout. Note that storage for disabled faces must be provided.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> </table>	Exists If:	[Surface Type] == 'SURFTYPE_CUBE'	Format:	Enable	Programming Notes	When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).
Exists If:	[Surface Type] == 'SURFTYPE_CUBE'						
Format:	Enable						
Programming Notes							
When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).							
2	<p>Cube Face Enable - Positive Y</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine: This field enable the individual face of a cube map. Enabling a face indicates that the face is present in the cube map, while disabling it indicates that that face is represented by the texture map's border color. Refer to Memory Data Formats for the correlation between faces and the cube map memory layout. Note that storage for disabled faces must be provided.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> </table>	Exists If:	[Surface Type] == 'SURFTYPE_CUBE'	Format:	Enable	Programming Notes	When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).
Exists If:	[Surface Type] == 'SURFTYPE_CUBE'						
Format:	Enable						
Programming Notes							
When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).							
1	<p>Cube Face Enable - Negative Z</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine: This field enable the individual face of a cube map. Enabling a face indicates that the face is present in the cube map, while disabling it indicates that that face is represented by the texture map's border color. Refer to Memory Data Formats for the correlation between faces and the cube map memory layout. Note that storage for disabled faces must be provided.</p>	Exists If:	[Surface Type] == 'SURFTYPE_CUBE'	Format:	Enable		
Exists If:	[Surface Type] == 'SURFTYPE_CUBE'						
Format:	Enable						

RENDER_SURFACE_STATE											
	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> </table>	Programming Notes		When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).							
Programming Notes											
When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).											
	<table border="1" style="width: 100%;"> <tr> <td style="width: 50px; vertical-align: top;">5:0</td> <td> Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] != 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> </td> </tr> </table>	5:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] != 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	[Surface Type] != 'SURFTYPE_CUBE'	Format:	MBZ				
5:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] != 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	[Surface Type] != 'SURFTYPE_CUBE'	Format:	MBZ						
Exists If:	[Surface Type] != 'SURFTYPE_CUBE'										
Format:	MBZ										
	<table border="1" style="width: 100%;"> <tr> <td style="width: 50px; vertical-align: top;">0</td> <td> Cube Face Enable - Positive Z <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine: This field enable the individual face of a cube map. Enabling a face indicates that the face is present in the cube map, while disabling it indicates that that face is represented by the texture map's border color. Refer to Memory Data Formats for the correlation between faces and the cube map memory layout. Note that storage for disabled faces must be provided.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> </table> </td> </tr> </table>	0	Cube Face Enable - Positive Z <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine: This field enable the individual face of a cube map. Enabling a face indicates that the face is present in the cube map, while disabling it indicates that that face is represented by the texture map's border color. Refer to Memory Data Formats for the correlation between faces and the cube map memory layout. Note that storage for disabled faces must be provided.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> </table>	Exists If:	[Surface Type] == 'SURFTYPE_CUBE'	Format:	Enable	Programming Notes		When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).	
0	Cube Face Enable - Positive Z <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Exists If:</td> <td>[Surface Type] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine: This field enable the individual face of a cube map. Enabling a face indicates that the face is present in the cube map, while disabling it indicates that that face is represented by the texture map's border color. Refer to Memory Data Formats for the correlation between faces and the cube map memory layout. Note that storage for disabled faces must be provided.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> </table>	Exists If:	[Surface Type] == 'SURFTYPE_CUBE'	Format:	Enable	Programming Notes		When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).			
Exists If:	[Surface Type] == 'SURFTYPE_CUBE'										
Format:	Enable										
Programming Notes											
When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).											
1	<table border="1" style="width: 100%;"> <tr> <td style="width: 50px; vertical-align: top;">31</td> <td> Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table> </td> </tr> </table>	31	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
31	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
	<table border="1" style="width: 100%;"> <tr> <td style="width: 50px; vertical-align: top;">30:24</td> <td> Memory Object Control State <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 this surface and the associated Auxiliary surface (if any).</p> </td> </tr> </table>	30:24	Memory Object Control State <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 this surface and the associated Auxiliary surface (if any).</p>	Format:	MEMORY_OBJECT_CONTROL_STATE						
30:24	Memory Object Control State <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 this surface and the associated Auxiliary surface (if any).</p>	Format:	MEMORY_OBJECT_CONTROL_STATE								
Format:	MEMORY_OBJECT_CONTROL_STATE										
	<table border="1" style="width: 100%;"> <tr> <td style="width: 50px; vertical-align: top;">23:19</td> <td> Base Mip Level <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>U4.1</td> </tr> </table> <p>Range: [0.0, 14.0]</p> <p>Specifies which mip level is considered the "base" level when determining mag-vs-min filter and selecting the "base" mip level.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">This field also exists in SAMPLER_STATE. If both fields are zero, the Base Mip Level is zero. If one is nonzero, Base Mip Level is the nonzero field. It is illegal to have both Base Mip Level fields nonzero.</td> </tr> </table> </td> </tr> </table>	23:19	Base Mip Level <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>U4.1</td> </tr> </table> <p>Range: [0.0, 14.0]</p> <p>Specifies which mip level is considered the "base" level when determining mag-vs-min filter and selecting the "base" mip level.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">This field also exists in SAMPLER_STATE. If both fields are zero, the Base Mip Level is zero. If one is nonzero, Base Mip Level is the nonzero field. It is illegal to have both Base Mip Level fields nonzero.</td> </tr> </table>	Format:	U4.1	Programming Notes		This field also exists in SAMPLER_STATE. If both fields are zero, the Base Mip Level is zero. If one is nonzero, Base Mip Level is the nonzero field. It is illegal to have both Base Mip Level fields nonzero.			
23:19	Base Mip Level <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>U4.1</td> </tr> </table> <p>Range: [0.0, 14.0]</p> <p>Specifies which mip level is considered the "base" level when determining mag-vs-min filter and selecting the "base" mip level.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">This field also exists in SAMPLER_STATE. If both fields are zero, the Base Mip Level is zero. If one is nonzero, Base Mip Level is the nonzero field. It is illegal to have both Base Mip Level fields nonzero.</td> </tr> </table>	Format:	U4.1	Programming Notes		This field also exists in SAMPLER_STATE. If both fields are zero, the Base Mip Level is zero. If one is nonzero, Base Mip Level is the nonzero field. It is illegal to have both Base Mip Level fields nonzero.					
Format:	U4.1										
Programming Notes											
This field also exists in SAMPLER_STATE. If both fields are zero, the Base Mip Level is zero. If one is nonzero, Base Mip Level is the nonzero field. It is illegal to have both Base Mip Level fields nonzero.											
	<table border="1" style="width: 100%;"> <tr> <td style="width: 50px; vertical-align: top;">18</td> <td> Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table> </td> </tr> </table>	18	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
18	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
	<table border="1" style="width: 100%;"> <tr> <td style="width: 50px; vertical-align: top;">17</td> <td> Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table> </td> </tr> </table>	17	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
17	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										

RENDER_SURFACE_STATE															
2	16	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													
	15	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													
14:0	Surface QPitch <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">QPitch[16:2]</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2"> 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. For Quilted Textures this field specifies the distance in rows between <i>quilt</i> slices. For compressed texture formats, one row contains a complete compression block vertically. SURFTYPE_3D: distance in <i>rows</i> between R-slices [Note: these <i>rows</i> are only in the vertical dimension without considering the depth dimension]. For compressed texture formats, one row contains a complete compression block vertically. Other surface types: field is ignored </td> </tr> </tbody> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">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> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td> For Surface Type 1D: This field must be set to an integer multiple of the Surface Horizontal Alignment For Surface Type 2D, CUBE: This field must be set to an integer multiple of the Surface Vertical Alignment For Surface Type 3D: <i>Tile Mode != Linear:</i> This field must be set to an integer multiple of the tile height (2^{Cv}) <i>Tile Mode == Linear:</i> This field must be set to an integer multiple of the Surface Vertical Alignment Note: for compressed textures (BC*, FXT1, ETC*, EAC*), this field is in units of rows of compression blocks. Note: for the compressed texture ASTC Surface Format, this field is in units of rows of compression blocks. 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. </td> </tr> </tbody> </table>	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. For Quilted Textures this field specifies the distance in rows between <i>quilt</i> slices. For compressed texture formats, one row contains a complete compression block vertically. SURFTYPE_3D: distance in <i>rows</i> between R-slices [Note: these <i>rows</i> are only in the vertical dimension without considering the depth dimension]. For compressed texture formats, one row contains a complete compression block vertically. Other surface types: field is ignored 		Value	Name	Description	[4h,1FFFCh]		in multiples of 4 (low 2 bits missing)	Programming Notes	For Surface Type 1D: This field must be set to an integer multiple of the Surface Horizontal Alignment For Surface Type 2D, CUBE: This field must be set to an integer multiple of the Surface Vertical Alignment For Surface Type 3D: <i>Tile Mode != Linear:</i> This field must be set to an integer multiple of the tile height (2^{Cv}) <i>Tile Mode == Linear:</i> This field must be set to an integer multiple of the Surface Vertical Alignment Note: for compressed textures (BC*, FXT1, ETC*, EAC*), this field is in units of rows of compression blocks. Note: for the compressed texture ASTC Surface Format, this field is in units of rows of compression blocks. 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.
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. For Quilted Textures this field specifies the distance in rows between <i>quilt</i> slices. For compressed texture formats, one row contains a complete compression block vertically. SURFTYPE_3D: distance in <i>rows</i> between R-slices [Note: these <i>rows</i> are only in the vertical dimension without considering the depth dimension]. For compressed texture formats, one row contains a complete compression block vertically. Other surface types: field is ignored 															
Value	Name	Description													
[4h,1FFFCh]		in multiples of 4 (low 2 bits missing)													
Programming Notes															
For Surface Type 1D: This field must be set to an integer multiple of the Surface Horizontal Alignment For Surface Type 2D, CUBE: This field must be set to an integer multiple of the Surface Vertical Alignment For Surface Type 3D: <i>Tile Mode != Linear:</i> This field must be set to an integer multiple of the tile height (2^{Cv}) <i>Tile Mode == Linear:</i> This field must be set to an integer multiple of the Surface Vertical Alignment Note: for compressed textures (BC*, FXT1, ETC*, EAC*), this field is in units of rows of compression blocks. Note: for the compressed texture ASTC Surface Format, this field is in units of rows of compression blocks. 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.															
31	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														
30	Reserved														

RENDER_SURFACE_STATE																											
29:16	<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, minus 1. If the surface is MIP-mapped, this field contains the height of the base MIP level. For buffers, this field specifies a portion of the buffer size.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 40%;">Description</th> <th style="width: 35%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,0]</td> <td></td> <td>must be zero</td> <td>[Surface Type] == 'SURFTYPE_1D'</td> </tr> <tr> <td>[0,16383]</td> <td></td> <td>height of surface - 1 (y/v dimension)</td> <td>[SurfaceType] == 'SURFTYPE_2D'</td> </tr> <tr> <td>[0,16383]</td> <td></td> <td>height of surface - 1 (y/v dimension)</td> <td>[SurfaceType] == 'SURFTYPE_3D'</td> </tr> <tr> <td>[0,16383]</td> <td></td> <td>height of surface - 1 (y/v dimension)</td> <td>[SurfaceType] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>[0,16383]</td> <td></td> <td>contains bits [20:7] of the number of entries in the buffer - 1</td> <td>([SurfaceType] == 'SURFTYPE_BUFFER') ([SurfaceType] == 'SURFTYPE_STRBUF')</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>For typed buffer and structured buffer surfaces, the number of entries in the buffer ranges from 1 to 2²⁷. For raw buffer surfaces, the number of entries in the buffer is the number of bytes which can range from 1 to 2³⁰. After subtracting one from the number of entries, software must place the fields of the resulting 27-bit value into the Height, Width, and Depth fields as indicated, right-justified in each field. Unused upper bits must be set to zero.</p> <p>If Vertical Line Stride is 1, this field indicates the height of the field, not the height of the frame</p> <p>The Height of a render target must be the same as the Height of the other render targets and the depth buffer (defined in 3DSTATE_DEPTH_BUFFER), unless Surface Type is SURFTYPE_1D or SURFTYPE_2D with Depth = 0 (non-array) and LOD = 0 (non-mip mapped).</p> <p>If this surface in memory is accessed with Vertical Line Stride set to both 0 and 1, this field must be an even value when Vertical Line Stride is 0.</p> <p>If Media Pixel Boundary Mode is not set to NORMAL_MODE, this field must be an even value.</p> <p>If Surface Format is PLANAR*, this field must be a multiple of 4</p>	Format:	U14-1	Value	Name	Description	Exists If	[0,0]		must be zero	[Surface Type] == 'SURFTYPE_1D'	[0,16383]		height of surface - 1 (y/v dimension)	[SurfaceType] == 'SURFTYPE_2D'	[0,16383]		height of surface - 1 (y/v dimension)	[SurfaceType] == 'SURFTYPE_3D'	[0,16383]		height of surface - 1 (y/v dimension)	[SurfaceType] == 'SURFTYPE_CUBE'	[0,16383]		contains bits [20:7] of the number of entries in the buffer - 1	([SurfaceType] == 'SURFTYPE_BUFFER') ([SurfaceType] == 'SURFTYPE_STRBUF')
Format:	U14-1																										
Value	Name	Description	Exists If																								
[0,0]		must be zero	[Surface Type] == 'SURFTYPE_1D'																								
[0,16383]		height of surface - 1 (y/v dimension)	[SurfaceType] == 'SURFTYPE_2D'																								
[0,16383]		height of surface - 1 (y/v dimension)	[SurfaceType] == 'SURFTYPE_3D'																								
[0,16383]		height of surface - 1 (y/v dimension)	[SurfaceType] == 'SURFTYPE_CUBE'																								
[0,16383]		contains bits [20:7] of the number of entries in the buffer - 1	([SurfaceType] == 'SURFTYPE_BUFFER') ([SurfaceType] == 'SURFTYPE_STRBUF')																								
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																										

RENDER_SURFACE_STATE

13:0

Width

Format:	U14-1
---------	-------

Description

This field specifies the width of the surface, minus 1. If the surface is MIP-mapped, this field specifies the width of the base MIP level. The width is specified in units of pixels or texels. For buffers, this field specifies a portion of the buffer size.

For surfaces accessed with the Media Block Read/Write message, this field is in units of DWords.

For surfaces accessed with the Transpose Read Message, this field is in units of DWords.

Value	Name	Description	Exists If
[0,16383]		width of surface - 1 (x/u dimension)	[SurfaceType] == 'SURFTYPE_1D'
[0,16383]		width of surface - 1 (x/u dimension)	[SurfaceType] == 'SURFTYPE_2D'
[0,16383]		width of surface - 1 (x/u dimension)	[SurfaceType] == 'SURFTYPE_3D'
[0,16383]		width of surface - 1 (x/u dimension)	[SurfaceType] == 'SURFTYPE_CUBE'
[0,127]		contains bits [6:0] of the number of entries in the buffer - 1	(([SurfaceType] == 'SURFTYPE_BUFFER') ([SurfaceType] == 'SURFTYPE_STRBUF'))

Programming Notes

- For surface types other than SURFTYPE_BUFFER or STRBUF 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 the Height.
- For MONO8 textures, Width must be a multiple of 32 texels.
- The **Width** of a render target must be the same as the **Width** of the other render target(s) and the depth buffer (defined in 3DSTATE_DEPTH_BUFFER), unless **Surface Type** is SURFTYPE_1D or SURFTYPE_2D with **Depth** = 0 (non-array) and **LOD** = 0 (non-mip mapped).
- The **Width** of a render target with YUV surface format must be a multiple of 2.
- For SURFTYPE_BUFFER: The low two bits of this field must be 11 if the Surface Format is RAW (the size of the buffer must be a multiple of 4 bytes).

If **Surface Format** is PLANAR*, this field must be a multiple of 2

A known issue exists if a primitive is rendered to the first 2 rows and last 2 columns of a 16K width surface. If any geometry is drawn inside this square it will be copied to column X=2 and X=3 (arrangement on Y position will stay the same). If any geometry exceeds the boundaries of this 2x2 region it will be drawn normally. The issue also only occurs if the surface has TileMode != Linear

RENDER_SURFACE_STATE																																													
3	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">31:21</td> <td> <p>Depth</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U11-1</td> </tr> </table> <p>This field specifies the total number of levels, minus 1, for a volume texture or the number of array elements, minus 1, 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. For buffers, this field specifies a portion of the buffer size.</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: 30%;">Description</th> <th style="width: 40%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,2047]</td> <td></td> <td>number of array elements - 1</td> <td>[SurfaceType] == 'SURFTYPE_1D'</td> </tr> <tr> <td>[0,2047]</td> <td></td> <td>number of array elements - 1</td> <td>[SurfaceType] == 'SURFTYPE_2D'</td> </tr> <tr> <td>[0,2047]</td> <td></td> <td>depth of surface - 1 (z/r dimension)</td> <td>[SurfaceType] == 'SURFTYPE_3D'</td> </tr> <tr> <td>[0,340]</td> <td></td> <td>number of array elements - 1 [see programming notes for range]</td> <td>[SurfaceType] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>[0,2047]</td> <td></td> <td>contains bits [31:21] of the number of entries in the buffer - 1</td> <td>(([SurfaceType] == SURFTYPE_BUFFER) OR ([SurfaceType] == 'SURFTYPE_STRBUF'))</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>The Depth of a render target must be the same as the Depth of the other render target(s) and of the depth buffer (defined in 3DSTATE_DEPTH_BUFFER).</p> <p>For SURFTYPE_CUBE: For Sampling Engine Surfaces and Typed Data Port Surfaces, the range of this field is [0,340], indicating the number of cube array elements (equal to the number of underlying 2D array elements divided by 6). For other surfaces, this field must be zero.</p> <p>For SURFTYPE_1D, 2D, and CUBE: The range of this field is reduced by one for each increase from zero of Minimum Array Element. For example, if Minimum Array Element is set to 1024 on a 2D surface, the range of this field is reduced to [0,1023].</p> </td> </tr> <tr> <td style="text-align: center;">20</td> <td> <p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> </td> </tr> <tr> <td style="text-align: center;">19</td> <td> <p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> </td> </tr> <tr> <td style="text-align: center;">18</td> <td> <p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table> </td> </tr> <tr> <td style="text-align: center;">17:0</td> <td> <p>Surface Pitch</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>U18-1 Pitch in #Bytes</td> </tr> </table> <p>Surface Pitch Range:</p> <ul style="list-style-type: none"> • For surfaces of type SURFTYPE_BUFFER: [0,2047] -> [1B, 2048B] • For surfaces of type SURFTYPE_STRBUF: [0,2047] -> [1B, 2048B] • For other linear surfaces: [0, 262143] -> [1B, 256KB] • For X-tiled surface: [511, 262143] -> [512B, 256KB] = [1 tile, 512 tiles] • For Y-tiled surfaces: [127, 262143]->[128B, 256KB] = [1 tile, 2048 tiles] </td> </tr> </table>	31:21	<p>Depth</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U11-1</td> </tr> </table> <p>This field specifies the total number of levels, minus 1, for a volume texture or the number of array elements, minus 1, 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. For buffers, this field specifies a portion of the buffer size.</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: 30%;">Description</th> <th style="width: 40%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,2047]</td> <td></td> <td>number of array elements - 1</td> <td>[SurfaceType] == 'SURFTYPE_1D'</td> </tr> <tr> <td>[0,2047]</td> <td></td> <td>number of array elements - 1</td> <td>[SurfaceType] == 'SURFTYPE_2D'</td> </tr> <tr> <td>[0,2047]</td> <td></td> <td>depth of surface - 1 (z/r dimension)</td> <td>[SurfaceType] == 'SURFTYPE_3D'</td> </tr> <tr> <td>[0,340]</td> <td></td> <td>number of array elements - 1 [see programming notes for range]</td> <td>[SurfaceType] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>[0,2047]</td> <td></td> <td>contains bits [31:21] of the number of entries in the buffer - 1</td> <td>(([SurfaceType] == SURFTYPE_BUFFER) OR ([SurfaceType] == 'SURFTYPE_STRBUF'))</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>The Depth of a render target must be the same as the Depth of the other render target(s) and of the depth buffer (defined in 3DSTATE_DEPTH_BUFFER).</p> <p>For SURFTYPE_CUBE: For Sampling Engine Surfaces and Typed Data Port Surfaces, the range of this field is [0,340], indicating the number of cube array elements (equal to the number of underlying 2D array elements divided by 6). For other surfaces, this field must be zero.</p> <p>For SURFTYPE_1D, 2D, and CUBE: The range of this field is reduced by one for each increase from zero of Minimum Array Element. For example, if Minimum Array Element is set to 1024 on a 2D surface, the range of this field is reduced to [0,1023].</p>	Format:	U11-1	Value	Name	Description	Exists If	[0,2047]		number of array elements - 1	[SurfaceType] == 'SURFTYPE_1D'	[0,2047]		number of array elements - 1	[SurfaceType] == 'SURFTYPE_2D'	[0,2047]		depth of surface - 1 (z/r dimension)	[SurfaceType] == 'SURFTYPE_3D'	[0,340]		number of array elements - 1 [see programming notes for range]	[SurfaceType] == 'SURFTYPE_CUBE'	[0,2047]		contains bits [31:21] of the number of entries in the buffer - 1	(([SurfaceType] == SURFTYPE_BUFFER) OR ([SurfaceType] == 'SURFTYPE_STRBUF'))	20	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	19	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	18	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	17:0	<p>Surface Pitch</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>U18-1 Pitch in #Bytes</td> </tr> </table> <p>Surface Pitch Range:</p> <ul style="list-style-type: none"> • For surfaces of type SURFTYPE_BUFFER: [0,2047] -> [1B, 2048B] • For surfaces of type SURFTYPE_STRBUF: [0,2047] -> [1B, 2048B] • For other linear surfaces: [0, 262143] -> [1B, 256KB] • For X-tiled surface: [511, 262143] -> [512B, 256KB] = [1 tile, 512 tiles] • For Y-tiled surfaces: [127, 262143]->[128B, 256KB] = [1 tile, 2048 tiles] 	Format:	U18-1 Pitch in #Bytes
31:21	<p>Depth</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U11-1</td> </tr> </table> <p>This field specifies the total number of levels, minus 1, for a volume texture or the number of array elements, minus 1, 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. For buffers, this field specifies a portion of the buffer size.</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: 30%;">Description</th> <th style="width: 40%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,2047]</td> <td></td> <td>number of array elements - 1</td> <td>[SurfaceType] == 'SURFTYPE_1D'</td> </tr> <tr> <td>[0,2047]</td> <td></td> <td>number of array elements - 1</td> <td>[SurfaceType] == 'SURFTYPE_2D'</td> </tr> <tr> <td>[0,2047]</td> <td></td> <td>depth of surface - 1 (z/r dimension)</td> <td>[SurfaceType] == 'SURFTYPE_3D'</td> </tr> <tr> <td>[0,340]</td> <td></td> <td>number of array elements - 1 [see programming notes for range]</td> <td>[SurfaceType] == 'SURFTYPE_CUBE'</td> </tr> <tr> <td>[0,2047]</td> <td></td> <td>contains bits [31:21] of the number of entries in the buffer - 1</td> <td>(([SurfaceType] == SURFTYPE_BUFFER) OR ([SurfaceType] == 'SURFTYPE_STRBUF'))</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>The Depth of a render target must be the same as the Depth of the other render target(s) and of the depth buffer (defined in 3DSTATE_DEPTH_BUFFER).</p> <p>For SURFTYPE_CUBE: For Sampling Engine Surfaces and Typed Data Port Surfaces, the range of this field is [0,340], indicating the number of cube array elements (equal to the number of underlying 2D array elements divided by 6). For other surfaces, this field must be zero.</p> <p>For SURFTYPE_1D, 2D, and CUBE: The range of this field is reduced by one for each increase from zero of Minimum Array Element. For example, if Minimum Array Element is set to 1024 on a 2D surface, the range of this field is reduced to [0,1023].</p>	Format:	U11-1	Value	Name	Description	Exists If	[0,2047]		number of array elements - 1	[SurfaceType] == 'SURFTYPE_1D'	[0,2047]		number of array elements - 1	[SurfaceType] == 'SURFTYPE_2D'	[0,2047]		depth of surface - 1 (z/r dimension)	[SurfaceType] == 'SURFTYPE_3D'	[0,340]		number of array elements - 1 [see programming notes for range]	[SurfaceType] == 'SURFTYPE_CUBE'	[0,2047]		contains bits [31:21] of the number of entries in the buffer - 1	(([SurfaceType] == SURFTYPE_BUFFER) OR ([SurfaceType] == 'SURFTYPE_STRBUF'))																		
Format:	U11-1																																												
Value	Name	Description	Exists If																																										
[0,2047]		number of array elements - 1	[SurfaceType] == 'SURFTYPE_1D'																																										
[0,2047]		number of array elements - 1	[SurfaceType] == 'SURFTYPE_2D'																																										
[0,2047]		depth of surface - 1 (z/r dimension)	[SurfaceType] == 'SURFTYPE_3D'																																										
[0,340]		number of array elements - 1 [see programming notes for range]	[SurfaceType] == 'SURFTYPE_CUBE'																																										
[0,2047]		contains bits [31:21] of the number of entries in the buffer - 1	(([SurfaceType] == SURFTYPE_BUFFER) OR ([SurfaceType] == 'SURFTYPE_STRBUF'))																																										
20	<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																																												
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	<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																																												
17:0	<p>Surface Pitch</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>U18-1 Pitch in #Bytes</td> </tr> </table> <p>Surface Pitch Range:</p> <ul style="list-style-type: none"> • For surfaces of type SURFTYPE_BUFFER: [0,2047] -> [1B, 2048B] • For surfaces of type SURFTYPE_STRBUF: [0,2047] -> [1B, 2048B] • For other linear surfaces: [0, 262143] -> [1B, 256KB] • For X-tiled surface: [511, 262143] -> [512B, 256KB] = [1 tile, 512 tiles] • For Y-tiled surfaces: [127, 262143]->[128B, 256KB] = [1 tile, 2048 tiles] 	Format:	U18-1 Pitch in #Bytes																																										
Format:	U18-1 Pitch in #Bytes																																												

RENDER_SURFACE_STATE

- For W-tiled surfaces: [127, 262143]->[128B, 256KB] = [1 tile, 2048 tiles]
- For TileYF and TileYS surfaces, the range is dependent on the Cu parameter (refer to *Memory Data Formats* 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 tile, 256KB/(2^{Cu}) tiles]

This field specifies the surface pitch in (#Bytes - 1).

For surfaces of type SURFTYPE_BUFFER and SURFTYPE_STRBUF, this field indicates the size of the structure.

Programming Notes

- For linear *render target* surfaces and surfaces accessed with the typed data port messages, the pitch must be a multiple of the element size for non-YUV surface formats. Pitch must be a multiple of 2 * element size for YUV surface formats.
- For untyped data port messages, which are only supported with **Surface Type** SURFTYPE_BUFFER, the pitch is ignored and assumed to be 1 byte.
- For linear surfaces with **Surface Type** of SURFTYPE_STRBUF, the pitch must be a multiple of 4 bytes.
- For linear surfaces with **Surface Type** of SURFTYPE_BUFFER and **Surface Format** RAW, the pitch must be 1 byte.
- For other linear surfaces, the pitch can be any multiple of bytes.
- For tiled surfaces, the pitch must be a multiple of the tile width.

If the surface is a stencil buffer (and thus has **Tile Mode** set to TILEMODE_WMAJOR), 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).

- The width of a tile depends on the surface format if Tiled Resource Enable is enabled. Refer to the Tiled Resource Enable field to determine which sub-mode applies to the surface format in use, and determine the Cu parameter from the Surface Layout section. The tile width is equal to 2^{Cu} bytes.
- For surfaces of type SURFTYPE_1D, this field is ignored.

The following table indicates the maximum byte width, frame width, and pitch size allowed when memory compression is on.

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

RENDER_SURFACE_STATE						
			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
4	31	Reserved				
		Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'			
		Format:	MBZ			
	30:29	Render Target And Sample Unorm Rotation				
		Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'			
		Description				
		For Render Target Surfaces: This field specifies the rotation of this render target surface when being written to memory.				
		For sample_unorm Messages: This field specifies the rotation of the data returned by sampler for sample_unorm message.				
		For Other Surfaces: This field is ignored.				
		Value	Name	Description		
		0h	0DEG	No rotation (0 degrees)		
		1h	90DEG	Rotate by 90 degrees		
		2h	180DEG	Rotate by 180 degrees [for sample_unorm message]		
		3h	270DEG	Rotate by 270 degrees		
		Programming Notes				
		Programming Notes for Render Target Surfaces only				
		<ul style="list-style-type: none"> Rotation is not supported for render targets of any type other than simple, non-mip-mapped, non-array 2D surfaces. The surface must be using tiled with X major. Width and Height fields apply to the dimensions of the surface before rotation. For 90 and 270 degree rotated surfaces, the Height (rather than the Width) must be less than or equal to the Surface Pitch (specified in bytes). For 90 and 270 degree rotated surfaces, the actual Height and Width of the surface in pixels (not the field value which is decremented) must both be even. 				

RENDER_SURFACE_STATE												
	<p>Rotation is supported only for surfaces with the following surface formats: B5G6R5_UNORM, B5G6R5_UNORM_SRGB, R8G8B8A8_UNORM, R8G8B8A8_UNORM_SRGB, B8G8R8[A X]8_UNORM, B8G8R8[A X]8_UNORM_SRGB, B10G10R10[A X]2_UNORM, B10G10R10A2_UNORM_SRGB, R10G10B10A2_UNORM, R10G10B10A2_UNORM_SRGB, R16G16B16A16_FLOAT, R16G16B16X16_FLOAT</p>											
28:18	<p>Minimum Array Element</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Surface Type] != 'SURFTYPE_STRBUF'</td> </tr> <tr> <td>Format:</td> <td>U11</td> </tr> </table>	Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'	Format:	U11							
Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'											
Format:	U11											
17:7	<p>Render Target View Extent</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Surface Type] != 'SURFTYPE_STRBUF'</td> </tr> <tr> <td>Format:</td> <td>U11-1</td> </tr> </table> <p>Range [0,2047] to indicate extent of [1,2048]</p> <p>For Render Target and Typed Dataport 3D Surfaces: This field indicates the extent of the accessible 'R' coordinates minus 1 on the LOD currently being rendered to.</p> <p>For Render Target and Typed Dataport 1D and 2D Surfaces: This field must be set to the same value as the Depth field.</p> <p>For Other Surfaces: This field is ignored.</p>	Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'	Format:	U11-1							
Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'											
Format:	U11-1											
6	<p>Multisampled Surface Storage Format</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Surface Type] != 'SURFTYPE_STRBUF'</td> </tr> </table> <p>This field indicates the storage format of the multisampled surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>MSS</td> <td>Multisampled surface was/is rendered as a render target</td> </tr> <tr> <td>1h</td> <td>DEPTH_STENCIL</td> <td>Multisampled surface was rendered as a depth or stencil buffer</td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> All multisampled render target surfaces must have this field set to MSFMT_MSS IF this field is MSFMT_DEPTH_STENCIL, the only sampling engine messages allowed are "ld2dms", "resinfo", and "sampleinfo". This field is ignored if Number of Multisamples is MULTISAMPLECOUNT_1 	Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'	Value	Name	Description	0h	MSS	Multisampled surface was/is rendered as a render target	1h	DEPTH_STENCIL	Multisampled surface was rendered as a depth or stencil buffer
Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'											
Value	Name	Description										
0h	MSS	Multisampled surface was/is rendered as a render target										
1h	DEPTH_STENCIL	Multisampled surface was rendered as a depth or stencil buffer										
5:3	<p>Number of Multisamples</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Surface Type] != 'SURFTYPE_STRBUF'</td> </tr> </table> <p>This field indicates the number of multisamples on the surface.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>MULTISAMPLECOUNT_1</td> </tr> <tr> <td>1h</td> <td>MULTISAMPLECOUNT_2</td> </tr> </tbody> </table>	Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'	Value	Name	0h	MULTISAMPLECOUNT_1	1h	MULTISAMPLECOUNT_2			
Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'											
Value	Name											
0h	MULTISAMPLECOUNT_1											
1h	MULTISAMPLECOUNT_2											

RENDER_SURFACE_STATE										
		<table border="1"> <tr> <td>2h</td> <td>MULTISAMPLECOUNT_4</td> </tr> <tr> <td>3h</td> <td>MULTISAMPLECOUNT_8</td> </tr> <tr> <td>4h</td> <td>MULTISAMPLECOUNT_16</td> </tr> <tr> <td>5h-7h</td> <td>Reserved</td> </tr> </table>	2h	MULTISAMPLECOUNT_4	3h	MULTISAMPLECOUNT_8	4h	MULTISAMPLECOUNT_16	5h-7h	Reserved
2h	MULTISAMPLECOUNT_4									
3h	MULTISAMPLECOUNT_8									
4h	MULTISAMPLECOUNT_16									
5h-7h	Reserved									
		<p style="text-align: center;">Programming Notes</p> <p>If this field is any value other than MULTISAMPLECOUNT_1, the Surface Type must be SURFTYPE_2D This field must be set to MULTISAMPLECOUNT_1 unless the surface is a Sampling Engine surface or Render Target surface.</p>								
	31:0	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Surface Type] == 'SURFTYPE_STRBUF'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	[Surface Type] == 'SURFTYPE_STRBUF'	Format:	MBZ				
Exists If:	[Surface Type] == 'SURFTYPE_STRBUF'									
Format:	MBZ									
	2:0	<p>Multisample Position Palette Index</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Surface Type] != 'SURFTYPE_STRBUF'</td> </tr> </table> <p>This field indicates the index into the sample position palette that the multisampled surface is using. This field is only used as a return value for the sampleinfo message, and is otherwise not used by hardware.</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>[0,7]</td> <td></td> </tr> </tbody> </table>	Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'	Value	Name	[0,7]			
Exists If:	[Surface Type] != 'SURFTYPE_STRBUF'									
Value	Name									
[0,7]										
5	31:25	<p>X Offset</p> <table border="1"> <tr> <td>Format:</td> <td>PixelOffset[8:2]</td> </tr> </table> <p>This field specifies the horizontal offset in pixels from the Surface Base Address to the start (origin) of the surface.</p> <p>This field effectively loosens the alignment restrictions on the origin of tiled surfaces. Previously, tiled surface origin was (by definition) located at the base address, and thus needed to satisfy the 4KB base address alignment restriction. Now the origin can be specified at a finer (4-wide x 4-high pixel) resolution.</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>[0,508]</td> <td></td> <td>In multiples of 4 (low 2 bits missing)</td> </tr> </tbody> </table>	Format:	PixelOffset[8:2]	Value	Name	Description	[0,508]		In multiples of 4 (low 2 bits missing)
Format:	PixelOffset[8:2]									
Value	Name	Description								
[0,508]		In multiples of 4 (low 2 bits missing)								
		<p style="text-align: center;">Programming Notes</p> <ul style="list-style-type: none"> For linear surfaces, this field must be zero. For surfaces accessed with the <i>Data Port Media Block Read/Write</i> message, the pixel size is assumed to be 32 bits in width. For surfaces accessed with the Data Port Transpose Read message, the pixel size is assumed to be 32 bits in width. For Surface Format with other than 8, 16, 32, 64, or 128 bits per pixel, this field must be zero. If Render Target Rotation is set to other than RTROTATE_0DEG, this field must be zero. 								

RENDER_SURFACE_STATE												
	<ul style="list-style-type: none"> • If Surface Type not SURFTYPE_2D, this field must be zero. • If MIP Count is not zero, this field must be zero. • If Number of Multisamples is not MULTISAMPLECOUNT_1, this field must be zero. • If Surface Array is enabled, this field must be zero. • If Auxiliary Surface Mode is not AUX_NONE, this field must be zero. • If Surface Vertical Alignment is VALIGN_8, this field must be a multiple of 8. • For Surface Format with 8 bits per element, this field must be a multiple of 16. • For Surface Format with 16 bits per element, this field must be a multiple of 8. 											
	<ul style="list-style-type: none"> • If Tiled Resource Mode is not TRMODE_NONE, this field must be zero. 											
24	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
23:21	Y Offset <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>RowOffset[4:2]</td> </tr> </table> <p>This field specifies the vertical offset in rows from the Surface Base Address to the start of the surface. (See additional description in the X Offset field.)</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>[0,28]</td> <td></td> <td>In multiples of 4 (low two bits missing)</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> <ul style="list-style-type: none"> • For linear surfaces, this field must be zero. • For render targets in which the Render Target Array Index is not zero, this field must be zero. • For Surface Format with other than 8, 16, 32, 64, or 128 bits per pixel, this field must be zero. • If Render Target Rotation is set to other than RTROTATE_0DEG, this field must be zero. • If Surface Type not SURFTYPE_2D, this field must be zero. • If MIP Count is not zero, this field must be zero. • If Number of Multisamples is not MULTISAMPLECOUNT_1, this field must be zero. • If Surface Array is enabled, this field must be zero. • If Auxiliary Surface Mode is not AUX_NONE, this field must be zero. </td> </tr> <tr> <td> <ul style="list-style-type: none"> • If Tiled Resource Mode is not TRMODE_NONE, this field must be zero. </td> </tr> </tbody> </table> <p>This field must be zero if Surface Format is Planar and the U and V planes are half-pitch (e.g. YV12 format).</p>	Format:	RowOffset[4:2]	Value	Name	Description	[0,28]		In multiples of 4 (low two bits missing)	Programming Notes	<ul style="list-style-type: none"> • For linear surfaces, this field must be zero. • For render targets in which the Render Target Array Index is not zero, this field must be zero. • For Surface Format with other than 8, 16, 32, 64, or 128 bits per pixel, this field must be zero. • If Render Target Rotation is set to other than RTROTATE_0DEG, this field must be zero. • If Surface Type not SURFTYPE_2D, this field must be zero. • If MIP Count is not zero, this field must be zero. • If Number of Multisamples is not MULTISAMPLECOUNT_1, this field must be zero. • If Surface Array is enabled, this field must be zero. • If Auxiliary Surface Mode is not AUX_NONE, this field must be zero. 	<ul style="list-style-type: none"> • If Tiled Resource Mode is not TRMODE_NONE, this field must be zero.
Format:	RowOffset[4:2]											
Value	Name	Description										
[0,28]		In multiples of 4 (low two bits missing)										
Programming Notes												
<ul style="list-style-type: none"> • For linear surfaces, this field must be zero. • For render targets in which the Render Target Array Index is not zero, this field must be zero. • For Surface Format with other than 8, 16, 32, 64, or 128 bits per pixel, this field must be zero. • If Render Target Rotation is set to other than RTROTATE_0DEG, this field must be zero. • If Surface Type not SURFTYPE_2D, this field must be zero. • If MIP Count is not zero, this field must be zero. • If Number of Multisamples is not MULTISAMPLECOUNT_1, this field must be zero. • If Surface Array is enabled, this field must be zero. • If Auxiliary Surface Mode is not AUX_NONE, this field must be zero. 												
<ul style="list-style-type: none"> • If Tiled Resource Mode is not TRMODE_NONE, this field must be zero. 												
20	EWA Disable For Cube <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Disable</td> </tr> </table> <p>Specifies if EWA mode for LOD quality improvement needs to be disabled for cube maps.</p>	Format:	Disable									
Format:	Disable											

RENDER_SURFACE_STATE																															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Enable [Default]</td> <td>EWA is enabled for cube maps</td> </tr> <tr> <td>1h</td> <td>Disable</td> <td>EWA is disabled for cube maps</td> </tr> </tbody> </table>	Value	Name	Description	0h	Enable [Default]	EWA is enabled for cube maps	1h	Disable	EWA is disabled for cube maps																				
Value	Name	Description																													
0h	Enable [Default]	EWA is enabled for cube maps																													
1h	Disable	EWA is disabled for cube maps																													
		Programming Notes																													
		This field indicates if EWA mode for LOD quality improvement needs to be disabled for cube maps. By default EWA would be on for cube maps hence this field must be 0. If there is any spec violation seen with EWA on cube maps then this field must be set to 1 to disable EWA for cubes.																													
19:18	Tiled Resource Mode For Sampling Engine, Render Target, and Typed/Untyped Surfaces: This field specifies the tiled resource mode. For other surfaces: This field is ignored.																														
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> <th>Exists If</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NONE</td> <td>No tiled resource</td> <td></td> </tr> <tr> <td>1h</td> <td>4KB</td> <td>4KB tiled resources</td> <td>[SurfaceType] == 'SURFTYPE_1D'</td> </tr> <tr> <td>2h</td> <td>64KB</td> <td>64KB tiled resources</td> <td>[SurfaceType] == 'SURFTYPE_1D'</td> </tr> <tr> <td>1h</td> <td>TILEYF</td> <td>4KB tiled resources</td> <td>[SurfaceType] != 'SURFTYPE_1D'</td> </tr> <tr> <td>2h</td> <td>TILEYS</td> <td>64KB tiled resources</td> <td>[SurfaceType] != 'SURFTYPE_1D'</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> <td></td> </tr> </tbody> </table>	Value	Name	Description	Exists If	0h	NONE	No tiled resource		1h	4KB	4KB tiled resources	[SurfaceType] == 'SURFTYPE_1D'	2h	64KB	64KB tiled resources	[SurfaceType] == 'SURFTYPE_1D'	1h	TILEYF	4KB tiled resources	[SurfaceType] != 'SURFTYPE_1D'	2h	TILEYS	64KB tiled resources	[SurfaceType] != 'SURFTYPE_1D'	3h	Reserved			
Value	Name	Description	Exists If																												
0h	NONE	No tiled resource																													
1h	4KB	4KB tiled resources	[SurfaceType] == 'SURFTYPE_1D'																												
2h	64KB	64KB tiled resources	[SurfaceType] == 'SURFTYPE_1D'																												
1h	TILEYF	4KB tiled resources	[SurfaceType] != 'SURFTYPE_1D'																												
2h	TILEYS	64KB tiled resources	[SurfaceType] != 'SURFTYPE_1D'																												
3h	Reserved																														
		Programming Notes																													
		If Tile Mode is not set to TILEMODE_YMAJOR, this field must be set to TRMODE_NONE, unless the Surface Type is SURFTYPE_1D.																													
		If this field is not set to TRMODE_NONE, the Surface Format must be one with 8, 16, 32, 64, or 128 bits per element, or one of the compressed texture modes (BC*, ETC*, EAC*, ASTC*). Additionally, YCRCB* formats are supported and treated as 16 bits per element, and the PLANAR_420_8 format is support and treated as 8 bits per element on the Y plane and 16 bits per element on the UV plane (if Separate UV Plane Enable is disabled) or 8 bits per element on the U and V planes (if Separate UV Plane Enable is enabled).																													
		If this field is set to TRMODE_NONE, the surface cannot contain any null pages unless Surface Type is BUFFER or STRBUF. A BUFFER or STRBUF surface with null pages must have Surface Base Address and Surface Pitch set to an integer multiple of the element size, and Surface Format must be one with 8, 16, 32, 64, or 128 bits per element.																													
		If Surface Format is PLANAR, the surface cannot contain any null pages.																													
17:16	Reserved																														
	Format:	MBZ																													
15	Reserved																														
	Format:	MBZ																													
14	Coherency Type Specifies the type of coherency maintained for this surface.																														
	Value	Name	Description																												

RENDER_SURFACE_STATE								
	0h	GPU coherent	Surface memory is kept coherent with GPU threads using GPU read/write ordering rules. Surface memory is backed by system memory but is not kept coherent with CPU (LLC).					
	1h	IA coherent	Surface memory is kept coherent with CPU (LLC).					
Programming Notes								
			This field may optionally be 1 (IA coherent) for messages sent to SFID_DP_DC0 or SFID_DP_DC1 or SFID_DP_DC2. This field is typically set to 0 (GPU coherent) if the context is operating in a non-SVM legacy mode (for example, Ring Buffer or a Execlist using 32-bit Virtual Address Legacy Context PPGTT32).					
			If the surface is not IA coherent, then it is possible that data written to Null Tile address will be cached, and then a later read to that same address might return a non-zero value. If a value is cached for a Null Tile address, the data will be eventually be discarded when the cache line is evicted. If the surface is IA coherent, then the cache line never contains a non-zero value, and Null Tile reads always return zero.					
13:12	Reserved							
	Format:		MBZ					
11:8	Mip Tail Start LOD							
	Format:		U4 in LOD Units					
			For Sampling Engine, Render Target, and Typed Surfaces: This field indicates which LOD is the first one in the MIP tail if Tiled Resource 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.					
Programming Notes								
			This field must be zero if the Surface Format is MONO8					
			This field is ignored if Tiled Resource Mode is TRMODE_NONE.					
			If Tiled Resource 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.					
			If Tiled Resource Mode is not TRMODE_NONE, to disable the Mip Tail this field must be set to a mip that larger than those present in the surface (i.e. 15). This is recommended for non-mip-mapped surfaces.					
			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:					
	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/	TlleYS	1	128x256	128x128	64x128	64x64	32x64

RENDER_SURFACE_STATE																																																										
		<table border="1"> <tr> <td rowspan="4">CUBE</td> <td></td> <td>2</td> <td>128x128</td> <td>128x64</td> <td>64x64</td> <td>64x32</td> <td>32x32</td> </tr> <tr> <td></td> <td>4</td> <td>64x128</td> <td>64x64</td> <td>32x64</td> <td>32x32</td> <td>16x32</td> </tr> <tr> <td></td> <td>8</td> <td>64x64</td> <td>64x32</td> <td>32x32</td> <td>32x16</td> <td>16x16</td> </tr> <tr> <td></td> <td>16</td> <td>32x64</td> <td>32x32</td> <td>16x32</td> <td>16x16</td> <td>8x16</td> </tr> <tr> <td></td> <td>TileYF</td> <td>any</td> <td>32x64</td> <td>32x32</td> <td>16x32</td> <td>16x16</td> <td>8x16</td> </tr> <tr> <td rowspan="2">3D</td> <td>TileYS</td> <td>1</td> <td>32x32x32</td> <td>16x32x32</td> <td>16x32x16</td> <td>16x16x16</td> <td>8x16x16</td> </tr> <tr> <td>TileYF</td> <td>1</td> <td>16x8x16</td> <td>8x8x16</td> <td>8x8x8</td> <td>8x4x8</td> <td>4x4x8</td> </tr> </table>	CUBE		2	128x128	128x64	64x64	64x32	32x32		4	64x128	64x64	32x64	32x32	16x32		8	64x64	64x32	32x32	32x16	16x16		16	32x64	32x32	16x32	16x16	8x16		TileYF	any	32x64	32x32	16x32	16x16	8x16	3D	TileYS	1	32x32x32	16x32x32	16x32x16	16x16x16	8x16x16	TileYF	1	16x8x16	8x8x16	8x8x8	8x4x8	4x4x8				
CUBE		2		128x128	128x64	64x64	64x32	32x32																																																		
		4		64x128	64x64	32x64	32x32	16x32																																																		
		8		64x64	64x32	32x32	32x16	16x16																																																		
		16	32x64	32x32	16x32	16x16	8x16																																																			
	TileYF	any	32x64	32x32	16x32	16x16	8x16																																																			
3D	TileYS	1	32x32x32	16x32x32	16x32x16	16x16x16	8x16x16																																																			
	TileYF	1	16x8x16	8x8x16	8x8x8	8x4x8	4x4x8																																																			
7:4	Surface Min LOD																																																									
	Format:		U4 In LOD Units																																																							
	For Sampling Engine and Typed Surfaces:																																																									
	This field indicates the most detailed LOD that can be accessed as part of this surface. This field is added to the delivered LOD (<i>sample_l</i> , <i>ld</i> , or <i>resinfo</i> message types) before it is used to address the surface.																																																									
	For Other Surfaces:																																																									
	This field is ignored.																																																									
	Programming Notes																																																									
	This field must be zero if the Surface Format is MONO8																																																									
3:0	MIP Count / LOD																																																									
	Format:	Sampling Engine and Typed Surfaces: U4 in (LOD units - 1) Render Target Surfaces: U4 in LOD units																																																								
	Range	Sampling Engine and Typed Surfaces: [0,14] representing [1,15] MIP levels Render Target Surfaces: [0,14] representing LOD Other Surfaces: [0]																																																								
	For Sampling Engine and Typed Surfaces:																																																									
	This field indicates the number of MIP levels allowed to be accessed starting at Surface Min LOD , which must be less than or equal to the number of MIP levels actually stored in memory for this surface. For <i>sample*</i> messages, the mip map access is clamped to be between the mipmap specified by the integer bits of the Min LOD and the ceiling of the value specified here. For <i>ld*</i> messages, out-of-bounds behavior results for LODs outside of the range specified in this field.																																																									
	For Render Target Surfaces:																																																									
	This field defines the MIP level that is currently being rendered into. This is the absolute MIP level on the surface and is not relative to the Surface Min LOD field, which is ignored for render target surfaces.																																																									
	For Other Surfaces:																																																									
	This field is reserved : MBZ																																																									
	Programming Notes																																																									
	The LOD of a render target must be the same as the LOD of the other render target(s) and of the depth buffer (defined in 3DSTATE_DEPTH_BUFFER).																																																									

RENDER_SURFACE_STATE											
		For render targets with YUV surface formats, the LOD must be zero. For sampling engine surfaces with YCRCB* or PLANAR* surface format, MIP Count must be zero.									
6	31	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>([Surface Format] != 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Surface Format] != 'PLANAR')	Format:	MBZ					
	Exists If:	([Surface Format] != 'PLANAR')									
	Format:	MBZ									
	31	<p>Separate UV Plane Enable</p> <table border="1"> <tr> <td>Exists If:</td> <td>([Surface Format] == 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If enabled, this field indicates that the U and V are present as separate planes. If disabled, the UV data is interleaved on a single plane.</p> <p style="text-align: center;">Programming Notes</p> <p>This field must be disabled (separate UV planes are not supported).</p> <p>See the section "Planar Memory Organization" for a description of how the size and location of the chroma planes (U and V) are calculated.</p>	Exists If:	([Surface Format] == 'PLANAR')	Format:	Enable					
Exists If:	([Surface Format] == 'PLANAR')										
Format:	Enable										
30	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td>([Surface Format] == 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Surface Format] == 'PLANAR')	Format:	MBZ						
Exists If:	([Surface Format] == 'PLANAR')										
Format:	MBZ										
30:16	<p>Auxiliary Surface QPitch</p> <table border="1"> <tr> <td>Exists If:</td> <td>([Surface Format] != 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>QPitch[16:2]</td> </tr> </table> <p>This field specifies the distance in rows between array slices on the auxiliary surface.</p> <table border="1"> <thead> <tr> <th>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> <p style="text-align: center;">Programming Notes</p> <p>This field must be set to an integer multiple of the Surface Vertical Alignment</p> <p>Software must ensure that this field is set to a value sufficiently large such that the array slices in the auxiliary surface do not overlap. Refer to the Memory Data Formats section for information on how surfaces are stored in memory.</p> <p>For non-multisampled render target's CCS auxiliary surface, QPitch must be computed with Horizontal Alignment = 128 and Surface Vertical Alignment = 256. These alignments are only for CCS buffer and not for associated render target.</p>	Exists If:	([Surface Format] != 'PLANAR')	Format:	QPitch[16:2]	Value	Name	Description	[4h,1FFFCh]		in multiples of 4 (low 2 bits missing)
Exists If:	([Surface Format] != 'PLANAR')										
Format:	QPitch[16:2]										
Value	Name	Description									
[4h,1FFFCh]		in multiples of 4 (low 2 bits missing)									
29:16		<p>X Offset for U or UV Plane</p> <table border="1"> <tr> <td>Exists If:</td> <td>([Surface Format] == 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>U14</td> </tr> </table> <p>This field specifies the horizontal offset in pixels from the Surface Base Address to the start (origin) of the U plane or interleaved UV plane, depending on the setting of Separate UV Plane Enable.</p> <p style="text-align: center;">Programming Notes</p>	Exists If:	([Surface Format] == 'PLANAR')	Format:	U14					
	Exists If:	([Surface Format] == 'PLANAR')									
Format:	U14										

RENDER_SURFACE_STATE												
		<p>This field must be a multiple of 4 (bits 1:0 MBZ).</p> <p>If Tiled Resource Mode is enabled, this field must be a multiple of the tile width in pixels.</p> <p>Auxiliary Surface Mode is forced to AUX_NONE.</p>										
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>Exists If:</td> <td>(([Surface Format] == 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	(([Surface Format] == 'PLANAR')	Format:	MBZ						
Exists If:	(([Surface Format] == 'PLANAR')											
Format:	MBZ											
14:12	Reserved	<table border="1"> <tr> <td>Exists If:</td> <td>(([Surface Format] != 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	(([Surface Format] != 'PLANAR')	Format:	MBZ						
Exists If:	(([Surface Format] != 'PLANAR')											
Format:	MBZ											
11:3	Auxiliary Surface Pitch	<table border="1"> <tr> <td>Exists If:</td> <td>(([Surface Format] != 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>U9-1 Pitch in #Tiles</td> </tr> </table> <p>This field specifies the Auxiliary surface pitch in (#Tiles - 1).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0, 511]</td> <td></td> <td>-> [1 tile, 512 tiles]</td> </tr> </tbody> </table>	Exists If:	(([Surface Format] != 'PLANAR')	Format:	U9-1 Pitch in #Tiles	Value	Name	Description	[0, 511]		-> [1 tile, 512 tiles]
Exists If:	(([Surface Format] != 'PLANAR')											
Format:	U9-1 Pitch in #Tiles											
Value	Name	Description										
[0, 511]		-> [1 tile, 512 tiles]										
13:0	Y Offset for U or UV Plane	<table border="1"> <tr> <td>Exists If:</td> <td>(([Surface Format] == 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>U14</td> </tr> </table> <p>This field specifies the vertical offset in rows from the Surface Base Address to the start (origin) of the U plane or interleaved UV plane, depending on the setting of Separate UV Plane Enable.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>For surfaces where Surface Format = PLANAR* and Separate UV Plane is Enabled, the Y Offset must be programmed in multiples of half-rows. For example, for a surface where Y is physically followed by U and then V in memory, the Y Offset to U plane would be (2*Y-Height). For all other PLANAR YUV formats this is programmed in multiples of full rows.</td> </tr> <tr> <td>For all format besides PLANAR_420_* This field must be a multiple of 4 (bits 1:0 MBZ). For formats PLANAR_420_* with separate chroma planes (e.g. YV12) this field must be multiple of 4 if U plane is the first chroma plane after the Y (luma) plane. It can be a multiple of 2 if it is the second chroma plane in memory. For formats PLANAR_420_* with interleaved chroma planes (e.g. NV12) this field can be multiple of 2.</td> </tr> <tr> <td>If Tiled Resource Mode is enabled, this field must be a multiple of the tile height in rows.</td> </tr> </tbody> </table> <p>Auxiliary Surface Mode is forced to AUX_NONE.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Workaround</th> </tr> </thead> <tbody> <tr> <td>For formats PLANAR_420_* when this field is not a multiple of 4 the Out-of-Bounds Supression check must be disabled to avoid false out of bound detection.</td> </tr> </tbody> </table>	Exists If:	(([Surface Format] == 'PLANAR')	Format:	U14	Programming Notes	For surfaces where Surface Format = PLANAR* and Separate UV Plane is Enabled, the Y Offset must be programmed in multiples of half-rows . For example, for a surface where Y is physically followed by U and then V in memory, the Y Offset to U plane would be (2*Y-Height). For all other PLANAR YUV formats this is programmed in multiples of full rows.	For all format besides PLANAR_420_* This field must be a multiple of 4 (bits 1:0 MBZ). For formats PLANAR_420_* with separate chroma planes (e.g. YV12) this field must be multiple of 4 if U plane is the first chroma plane after the Y (luma) plane. It can be a multiple of 2 if it is the second chroma plane in memory. For formats PLANAR_420_* with interleaved chroma planes (e.g. NV12) this field can be multiple of 2.	If Tiled Resource Mode is enabled, this field must be a multiple of the tile height in rows.	Workaround	For formats PLANAR_420_* when this field is not a multiple of 4 the Out-of-Bounds Supression check must be disabled to avoid false out of bound detection.
Exists If:	(([Surface Format] == 'PLANAR')											
Format:	U14											
Programming Notes												
For surfaces where Surface Format = PLANAR* and Separate UV Plane is Enabled, the Y Offset must be programmed in multiples of half-rows . For example, for a surface where Y is physically followed by U and then V in memory, the Y Offset to U plane would be (2*Y-Height). For all other PLANAR YUV formats this is programmed in multiples of full rows.												
For all format besides PLANAR_420_* This field must be a multiple of 4 (bits 1:0 MBZ). For formats PLANAR_420_* with separate chroma planes (e.g. YV12) this field must be multiple of 4 if U plane is the first chroma plane after the Y (luma) plane. It can be a multiple of 2 if it is the second chroma plane in memory. For formats PLANAR_420_* with interleaved chroma planes (e.g. NV12) this field can be multiple of 2.												
If Tiled Resource Mode is enabled, this field must be a multiple of the tile height in rows.												
Workaround												
For formats PLANAR_420_* when this field is not a multiple of 4 the Out-of-Bounds Supression check must be disabled to avoid false out of bound detection.												

RENDER_SURFACE_STATE

2:0

Auxiliary Surface Mode

Exists If:	([Surface Format] != 'PLANAR')
Format:	U3

Specifies what type of surface the Auxiliary surface is. The Auxiliary surface has its own base address and pitch, but otherwise shares or overrides other fields set for the primary surface, detailed in the programming notes below.

Value	Name	Description
0h	AUX_NONE	No Auxiliary surface is used
1h	AUX_CCS_D	The Auxiliary surface is a CCS (Color Control Surface) with compression disabled or an MCS with compression enabled, depending on Number of Multisamples . MCS (Multisample Control Surface) is a special type of CCS.
2h	AUX_APPEND	The Auxiliary surface is an append buffer
3h	AUX_HIZ	The Auxiliary surface is a hierarchical depth buffer
4h	Reserved	
5h	AUX_CCS_E	The Auxiliary surface is a CCS with compression enabled or an MCS with compression enabled, depending on Number of Multisamples .
6h-7h	Reserved	

Programming Notes

The CCS and hierarchical depth Auxiliary surface shares **Height, Width, Depth, Surface Type, Surface Array, Surface Min LOD, MIP Count / LOD, Surface Object Control State, Resource Min LOD, and Minimum Array Element** with the primary surface. The hierarchical depth Auxiliary surface uses **Surface Horizontal Alignment** of 16, **Surface Vertical Alignment** of 8, regardless of the primary surface's values for these fields. **X & Y Offset** are set to zero for the purpose of accessing the Auxiliary surface. If this field is set to AUX_HIZ, **Surface Format** must be one of the following: R32_FLOAT, R24_UNORM_X8_TYPELESS, or R16_UNORM, and the format must match the format used when the surface was used as a depth buffer (with R channel corresponding to D channel).

CCS and hierarchical depth Auxiliary surfaces are TileY with **Tiled Resource Mode** of TRMODE_NONE regardless of the tile mode of the primary surface, and **Mip Tail Start LOD** is ignored for these surfaces.

The CCS Auxiliary surface for non-multisampled render targets has Horizontal Alignment = 128 and Vertical alignment = 64.

The CCS Auxiliary surface for **Number of Multisamples** > 1 uses **Surface Horizontal Alignment** of 16 and **Surface Vertical Alignment** of 4 regardless of the primary surface's values for these fields.

If this field is set to AUX_HIZ, **Number of Multisamples** must be MULTISAMPLECOUNT_1, and Surface Type cannot be SURFTYPE_3D.

If **Number of Multisamples** is MULTISAMPLECOUNT_1, AUX_CCS_E setting is only allowed if **Surface Format** is supported for Render Target Compression. This setting enables render target compression.

If **Number of Multisamples** is MULTISAMPLECOUNT_1, AUX_CCS_D setting is only allowed if **Surface Format** supported for Fast Clear. In addition, if the surface is bound to the sampling engine, **Surface Format** must be supported for Render Target Compression for surfaces bound to the sampling engine. For render target surfaces, this setting disables render target compression. For sampling engine surfaces, this mode behaves the same as AUX_CCS_E.

If **Number of Multisamples** is MULTISAMPLECOUNT_1 and AUX_CCS_E or AUX_CCS_D

RENDER_SURFACE_STATE											
7	31	<p>Memory Compression Mode Distinguishes Vertical from Horizontal compression.</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>Horizontal [Default]</td> <td></td> </tr> <tr> <td style="text-align: center;">1</td> <td>Vertical</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0	Horizontal [Default]		1	Vertical	
	Value	Name	Description								
	0	Horizontal [Default]									
	1	Vertical									
	30	<p>Memory Compression Enable</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This surface may contain compressed or compressible pixels. Memory compression will be attempted for writes to this surface. Reads from this surface will check for compressed data.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; background-color: #e1eef6;">Programming Notes</th> </tr> <tr> <td>The compression control must have 0 value for non-tileY modes. The Memory Compression Enable can be non-zero only for the surface state that has media messages. That is for 3d case the compression control bits will be 0 in normal surface state but can be non-zero in normal surface state for media messages. E.g. <i>sample_unorm</i>.</td> </tr> <tr> <td>The only sampler messages supported with memory compression enabled are <i>sample_8x8</i>, <i>sample_unorm</i>, and SIMD16 <i>sample</i>.</td> </tr> <tr> <td>Please refer to vol1a Memory Data Formats chapter Media Memory Compression for more details, including format restrictions.</td> </tr> </table>	Format:	Enable	Programming Notes	The compression control must have 0 value for non-tileY modes. The Memory Compression Enable can be non-zero only for the surface state that has media messages. That is for 3d case the compression control bits will be 0 in normal surface state but can be non-zero in normal surface state for media messages. E.g. <i>sample_unorm</i> .	The only sampler messages supported with memory compression enabled are <i>sample_8x8</i> , <i>sample_unorm</i> , and SIMD16 <i>sample</i> .	Please refer to vol1a Memory Data Formats chapter Media Memory Compression for more details, including format restrictions.			
	Format:	Enable									
	Programming Notes										
	The compression control must have 0 value for non-tileY modes. The Memory Compression Enable can be non-zero only for the surface state that has media messages. That is for 3d case the compression control bits will be 0 in normal surface state but can be non-zero in normal surface state for media messages. E.g. <i>sample_unorm</i> .										
	The only sampler messages supported with memory compression enabled are <i>sample_8x8</i> , <i>sample_unorm</i> , and SIMD16 <i>sample</i> .										
	Please refer to vol1a Memory Data Formats chapter Media Memory Compression for more details, including format restrictions.										
29:28	<p>Reserved</p> <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										
27:25	<p>Shader Channel Select Red</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">Format:</td> <td>Shader Channel Select Enumerated Type</td> </tr> </table> <p>Specifies which surface channel is read or written in the Red shader channel.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; background-color: #e1eef6;">Programming Notes</th> </tr> <tr> <td>The Shader channel selects also define which shader channels are written to which surface channel. If the Shader channel select is SCS_ZERO or SCS_ONE then it is not written to the surface. If the shader channel select is SCS_RED it is written to the surface red channel and so on. If more than one shader channel select is set to the same surface channel only the first shader channel in RGBA order will be written. Each shader channel select must be set to the same surface channel (R = SCS_RED, G = SCS_GREEN, B = SCS_BLUE, A = SCS_ALPHA) if the surface is accessed via the sampler's <i>sample_unorm*</i> or <i>sample_8x8</i> messages.</td> </tr> <tr> <td>The Shader Channel Select fields do not affect the following sampling engine message types: <i>resinfo</i>, <i>sampleinfo</i>, <i>LOD</i>, and <i>ld_mcs</i>. These messages behave as if each Shader Channel Select is set to the same color surface channel.</td> </tr> <tr> <td>For the sampling engine <i>gather4*</i> messages, the Gather4 Source Channel Select field in the message header defines which channel's Shader Channel Select is used to select the surface channel to be sampled. Other Shader Channel Select fields are ignored.</td> </tr> <tr> <td>For the sampling engine <i>sample*_c</i> and <i>gather4*_c</i> messages, the compare operation always occurs on the red channel from the surface regardless of the setting of the Shader Channel Select fields.</td> </tr> <tr> <td>For Render Target, Red, Green and Blue Shader Channel Selects MUST be such that only valid</td> </tr> </table>	Format:	Shader Channel Select Enumerated Type	Programming Notes	The Shader channel selects also define which shader channels are written to which surface channel. If the Shader channel select is SCS_ZERO or SCS_ONE then it is not written to the surface. If the shader channel select is SCS_RED it is written to the surface red channel and so on. If more than one shader channel select is set to the same surface channel only the first shader channel in RGBA order will be written. Each shader channel select must be set to the same surface channel (R = SCS_RED, G = SCS_GREEN, B = SCS_BLUE, A = SCS_ALPHA) if the surface is accessed via the sampler's <i>sample_unorm*</i> or <i>sample_8x8</i> messages.	The Shader Channel Select fields do not affect the following sampling engine message types: <i>resinfo</i> , <i>sampleinfo</i> , <i>LOD</i> , and <i>ld_mcs</i> . These messages behave as if each Shader Channel Select is set to the same color surface channel.	For the sampling engine <i>gather4*</i> messages, the Gather4 Source Channel Select field in the message header defines which channel's Shader Channel Select is used to select the surface channel to be sampled. Other Shader Channel Select fields are ignored.	For the sampling engine <i>sample*_c</i> and <i>gather4*_c</i> messages, the compare operation always occurs on the red channel from the surface regardless of the setting of the Shader Channel Select fields.	For Render Target, Red, Green and Blue Shader Channel Selects MUST be such that only valid		
Format:	Shader Channel Select Enumerated Type										
Programming Notes											
The Shader channel selects also define which shader channels are written to which surface channel. If the Shader channel select is SCS_ZERO or SCS_ONE then it is not written to the surface. If the shader channel select is SCS_RED it is written to the surface red channel and so on. If more than one shader channel select is set to the same surface channel only the first shader channel in RGBA order will be written. Each shader channel select must be set to the same surface channel (R = SCS_RED, G = SCS_GREEN, B = SCS_BLUE, A = SCS_ALPHA) if the surface is accessed via the sampler's <i>sample_unorm*</i> or <i>sample_8x8</i> messages.											
The Shader Channel Select fields do not affect the following sampling engine message types: <i>resinfo</i> , <i>sampleinfo</i> , <i>LOD</i> , and <i>ld_mcs</i> . These messages behave as if each Shader Channel Select is set to the same color surface channel.											
For the sampling engine <i>gather4*</i> messages, the Gather4 Source Channel Select field in the message header defines which channel's Shader Channel Select is used to select the surface channel to be sampled. Other Shader Channel Select fields are ignored.											
For the sampling engine <i>sample*_c</i> and <i>gather4*_c</i> messages, the compare operation always occurs on the red channel from the surface regardless of the setting of the Shader Channel Select fields.											
For Render Target, Red, Green and Blue Shader Channel Selects MUST be such that only valid											

RENDER_SURFACE_STATE													
	<p>components can be swapped i.e. only change the order of components in the pixel. Any other values for these Shader Channel Select fields are not valid for Render Targets. This also means that there MUST not be multiple shader channels mapped to the same RT channel.</p> <p>When multiple Channel selects have the same value and shader channel is disabled, disable channel writes 0s to memory. This behavior does not match with Data Port message via HDC.</p> <p>The output channel is undefined if the source is to a channel is not present for the current surface format. For example, If the surface format is R16_float and the shader channel select green specifies green as the source the output is undefined. It should instead select 0 which is the default for a missing color channel..</p>												
24:22	<p>Shader Channel Select Green</p> <table border="1"> <tr> <td>Format:</td> <td>Shader Channel Select Enumerated Type</td> </tr> </table> <p>See Shader Channel Select Red for details.</p>	Format:	Shader Channel Select Enumerated Type										
Format:	Shader Channel Select Enumerated Type												
21:19	<p>Shader Channel Select Blue</p> <table border="1"> <tr> <td>Format:</td> <td>Shader Channel Select Enumerated Type</td> </tr> </table> <p>See Shader Channel Select Red for details.</p>	Format:	Shader Channel Select Enumerated Type										
Format:	Shader Channel Select Enumerated Type												
18:16	<p>Shader Channel Select Alpha</p> <table border="1"> <tr> <td>Format:</td> <td>Shader Channel Select Enumerated Type</td> </tr> </table> <p>See Shader Channel Select Red for details.</p> <table border="1" style="background-color: #e6f2ff;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For Render Target, this field MUST be programmed to value = SCS_ALPHA.</td> </tr> </table>	Format:	Shader Channel Select Enumerated Type	Programming Notes		For Render Target, this field MUST be programmed to value = SCS_ALPHA.							
Format:	Shader Channel Select Enumerated Type												
Programming Notes													
For Render Target, this field MUST be programmed to value = SCS_ALPHA.													
15:12	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
11:0	<p>Resource Min LOD</p> <table border="1"> <tr> <td>Format:</td> <td>U4.8 in LOD units</td> </tr> </table> <p>For Sampling Engine Surfaces: This field indicates the most detailed LOD that is present in the resource underlying the surface. Refer to the "LOD Computation Pseudocode" section for the use of this field.</p> <p>For Other Surfaces: This field is ignored.</p> <table border="1" style="background-color: #e6f2ff;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,14]</td> <td></td> </tr> </tbody> </table> <table border="1" style="background-color: #e6f2ff;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">This field must be zero if the Surface Format is MONO8</td> </tr> <tr> <td colspan="2">This field must be zero if the ChromaKey Enable is enabled in the associated sampler.</td> </tr> </table>	Format:	U4.8 in LOD units	Value	Name	[0,14]		Programming Notes		This field must be zero if the Surface Format is MONO8		This field must be zero if the ChromaKey Enable is enabled in the associated sampler.	
Format:	U4.8 in LOD units												
Value	Name												
[0,14]													
Programming Notes													
This field must be zero if the Surface Format is MONO8													
This field must be zero if the ChromaKey Enable is enabled in the associated sampler.													
8..9	<p>63:0 Surface Base Address</p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[63:0]SurfaceBase</td> </tr> </table> <p>Specifies the byte-aligned base address of the surface.</p>	Format:	GraphicsAddress[63:0]SurfaceBase										
Format:	GraphicsAddress[63:0]SurfaceBase												

RENDER_SURFACE_STATE

Programming Notes

- For SURFTYPE_BUFFER render targets, this field specifies the base address of first element of the surface. The surface is interpreted as a simple array of that single element type. The address must be naturally-aligned to the element size (e.g., a buffer containing R32G32B32A32_FLOAT elements must be 16-byte aligned).
- For SURFTYPE_BUFFER non-rendertarget surfaces, this field specifies the base address of the first element of the surface, computed in software by adding the surface base address to the byte offset of the element in the buffer. The base address must be aligned to element size.
- Linear depth buffer surface base addresses must be 64-byte aligned. Note that while render targets (color) can be SURFTYPE_BUFFER, depth buffers cannot.
- Mipmapped surfaces are stored in a "monolithic" (fixed) format, and only require a single address for the base MIP. All other MIPs are positioned relative to the base MIP.
- The Base Address for linear (non-tiled) render target surfaces and surfaces accessed with the typed surface read/write data port messages must be element-size aligned for Non-YUV surface formats, or a multiple of 2 element-sizes for YUV surface formats.
- Other linear (non-tiled) surfaces have no alignment requirements (byte alignment is sufficient).
- For tiled surfaces, the actual start of the surface can be offset from the Surface Base Address by the X Offset and Y Offset fields. Tiles are inherently page-aligned (4K or 64K).
- Certain message types used to access surfaces have more stringent alignment requirements. Please refer to the specific data-port message documentation for additional restrictions.

Tiled surface base addresses must be 4KB-aligned. Note that only the offsets from Surface Base Address are tiled, Surface Base Address itself is not transformed using the tiling algorithm.

Tiled surface base addresses must be tile aligned (64KB aligned for TileYS, 4KB aligned for all other tile modes). For 1D surfaces, the base address must be 64KB aligned if Tiled Resource Mode is TRMODE_64KB, and 4KB aligned if Tiled Resource Mode is TRMODE_4KB. An exception to this tile alignment is when a SURFACE_STATE describes a single MIP within the MIP Tail of another surface, using a 64-bit or 128-bit Surface Format —then Surface Base Address can refer directly to the given MIP (e.g. to write to a non-renderable Surface Format by re-describing as an alternative surface).

RENDER_SURFACE_STATE		
10..11	63:62	Reserved
		Exists If: ([Surface Format] == 'PLANAR')
		Format: MBZ
	61:48	X Offset for V Plane
		Exists If: ([Surface Format] == 'PLANAR')
		Format: U14
		This field specifies the horizontal offset in pixels from the Surface Base Address to the start (origin) of the V plane.
		Programming Notes
		This field must be a multiple of 4 (bits 1:0 MBZ).
		If Tiled Resource Mode is enabled, this field must be a multiple of the tile width in pixels.
	This field is ignored if Separate UV Plane Enable is disabled.	
47:46		Reserved
		Exists If: ([Surface Format] == 'PLANAR')
		Format: MBZ
45:32		Y Offset for V Plane
		Exists If: ([Surface Format] == 'PLANAR')
		Format: U14
		This field specifies the vertical offset in rows from the Surface Base Address to the start (origin) of the V plane.
		Programming Notes
		For surfaces where Surface Format = PLANAR* and Separate UV Plane is Enabled, the Y Offset must be programmed in multiples of half-rows . For example, for a surface where Y is physically followed by U and then V in memory, the Y Offset to V plane would be (2*Y-Height+ U-Height). For all other PLANAR YUV formats this is programmed in multiples of full rows (e.g Y-Height + U-Height).
		For all format besides PLANAR_420_* This field must be a multiple of 4 (bits 1:0 MBZ). For formats PLANAR_420_* this field must be multiple of 4 if U plane is the first chroma plane after the Y (luma) plane. It can be a multiple of 2 if it is the second chroma plane. For formats PLANAR_420_* when this field is not a multiple of 4 the Out-of-Bounds Suppression check must be disabled to avoid false out of bound detection.
		If Tiled Resource Mode is enabled, this field must be a multiple of the tile height in rows.
		This field is ignored if Separate UV Plane Enable is disabled.
	31:21	
		Exists If: [Memory Compression Enable] == 1
		This field is valid only if Media Memory Compression is on for the surface(Memory Compression Enable == 1). In that case, the Auxiliary Surface Base address is never expected to be used and hence can be overloaded. This represents the 11 bit index into the table in memory which maps the surface to the auxiliary base address.

RENDER_SURFACE_STATE										
63:12	Auxiliary Surface Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>([Surface Format] != 'PLANAR') AND [Memory Compression Enable] == 0</td> </tr> <tr> <td>Format:</td> <td>GraphicsAddress[63:12]</td> </tr> </table> <p>Specifies the 4kbyte-aligned base address of the Auxiliary surface associated with the primary surface specified in other SURFACE_STATE fields.</p>		Exists If:	([Surface Format] != 'PLANAR') AND [Memory Compression Enable] == 0	Format:	GraphicsAddress[63:12]				
Exists If:	([Surface Format] != 'PLANAR') AND [Memory Compression Enable] == 0									
Format:	GraphicsAddress[63:12]									
11	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ						
Format:	MBZ									
10	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>MBZ</td> </tr> </table>		Format:	MBZ						
Format:	MBZ									
9:5	Quilt Height <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>U5</td> </tr> </table> <p>This field specifies the height of a quilted texture in units of quilt slices. Refer to the section on Quilted Textures for more details.</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>[0,31]</td> <td></td> <td>representing height of quilt - 1 (y/v dimension)</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>Programming Notes</p> <ul style="list-style-type: none"> Only power-of-2 Quilt Height and Quilt Width values are allowed: (1,2,4,8,16,32) mapping to (0,1,3,7,15,31) values in the fields. A surface is defined as a quilted texture if either Quilt Height or Quilt Width is nonzero (actual field value, not the incremented value). A quilted texture <ul style="list-style-type: none"> is only supported by the sampling engine (other shared functions will ignore the Quilt Width and Quilt Height field, behaving as if they are set to zero). must have a Surface Type of SURFTYPE_2D. must have Number of Multisamples set to NUMSAMPLES_1. must have Vertical Line Stride set to 0. must have Auxiliary Surface Mode set to AUX_NONE. Depth indicates the array dimension of the quilted texture if Surface Array is enabled. The valid range of Depth is [0, 2048 / (QuiltWidth * QuiltHeight) - 1], i.e. the total number of underlying array slices including quilt slices cannot exceed 2048. cannot be accessed with any Id* message type or using a sampler with the Non-Normalized Coordinate Enable field enabled. </div>		Format:	U5	Value	Name	Description	[0,31]		representing height of quilt - 1 (y/v dimension)
Format:	U5									
Value	Name	Description								
[0,31]		representing height of quilt - 1 (y/v dimension)								

RENDER_SURFACE_STATE												
	4:0	<p>Quilt Width</p> <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>This field specifies the width of a quilted texture in units of quilt slices. Refer to the section on Quilted Textures for more details.</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>representing width of quilt - 1 (x/u dimension)</td> </tr> </tbody> </table>	Format:	U5	Value	Name	Description	[0,31]		representing width of quilt - 1 (x/u dimension)		
	Format:	U5										
Value	Name	Description										
[0,31]		representing width of quilt - 1 (x/u dimension)										
12	31:0	<p>Reserved</p> <table border="1"> <tr> <td>Exists If:</td> <td> ([Auxiliary Surface Mode] != 'AUX_CCS_D') AND ([Auxiliary Surface Mode] != 'AUX_CCS_E') AND ([Auxiliary Surface Mode] != 'AUX_HIZ') </td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Auxiliary Surface Mode] != 'AUX_CCS_D') AND ([Auxiliary Surface Mode] != 'AUX_CCS_E') AND ([Auxiliary Surface Mode] != 'AUX_HIZ')	Format:	MBZ						
	Exists If:	([Auxiliary Surface Mode] != 'AUX_CCS_D') AND ([Auxiliary Surface Mode] != 'AUX_CCS_E') AND ([Auxiliary Surface Mode] != 'AUX_HIZ')										
	Format:	MBZ										
	31:0	<p>Red Clear Color</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'</td> </tr> <tr> <td>Format:</td> <td>S31 (2's complement) for all SINT surface formats</td> </tr> <tr> <td>Format:</td> <td>U32 for all UINT surface formats</td> </tr> <tr> <td>Format:</td> <td>IEEE_FP for all other surface formats</td> </tr> </table> <p>For Sampling Engine Surfaces and Render Targets with Auxiliary Surface Mode set to AUX_CCS: Specifies the clear value for the red channel. For Other Surfaces: This field is ignored.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>If Number of Multisamples is not MULTISAMPLECOUNT_1, only 0/1 values allowed</td> </tr> <tr> <td>If Number of Multisamples is MULTISAMPLECOUNT_1 AND if this RT is fast cleared with non-0/1 clear value, this RT must be partially resolved (refer to Partial Resolve operation) before binding this surface to Sampler.</td> </tr> </tbody> </table>	Exists If:	[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'	Format:	S31 (2's complement) for all SINT surface formats	Format:	U32 for all UINT surface formats	Format:	IEEE_FP for all other surface formats	Programming Notes	If Number of Multisamples is not MULTISAMPLECOUNT_1, only 0/1 values allowed
Exists If:	[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'											
Format:	S31 (2's complement) for all SINT surface formats											
Format:	U32 for all UINT surface formats											
Format:	IEEE_FP for all other surface formats											
Programming Notes												
If Number of Multisamples is not MULTISAMPLECOUNT_1, only 0/1 values allowed												
If Number of Multisamples is MULTISAMPLECOUNT_1 AND if this RT is fast cleared with non-0/1 clear value, this RT must be partially resolved (refer to Partial Resolve operation) before binding this surface to Sampler.												
31:0	<p>Hierarchical Depth Clear Value</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Auxiliary Surface Mode] == 'AUX_HIZ'</td> </tr> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table> <p>If Auxiliary Surface Mode is AUX_HIZ, this field specifies the depth clear value associated with this surface. If disabled, this field is ignored.</p>	Exists If:	[Auxiliary Surface Mode] == 'AUX_HIZ'	Format:	IEEE_Float							
Exists If:	[Auxiliary Surface Mode] == 'AUX_HIZ'											
Format:	IEEE_Float											
13	31:0	<p>Green Clear Color</p> <table border="1"> <tr> <td>Exists If:</td> <td>[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'</td> </tr> <tr> <td>Format:</td> <td>S31 (2's complement) for all SINT surface formats</td> </tr> <tr> <td>Format:</td> <td>U32 for all UINT surface formats</td> </tr> <tr> <td>Format:</td> <td>IEEE_FP for all other surface formats</td> </tr> </table> <p>For Sampling Engine Surfaces and Render Targets with Auxiliary Surface Mode set to AUX_CCS: Specifies the clear value for the green channel. For Other Surfaces: This field is ignored.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>If programmed to non 0/1 values, SW must ensure a render target partial resolve pass before binding a cleared RT to texture.</td> </tr> </tbody> </table>	Exists If:	[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'	Format:	S31 (2's complement) for all SINT surface formats	Format:	U32 for all UINT surface formats	Format:	IEEE_FP for all other surface formats	Programming Notes	If programmed to non 0/1 values, SW must ensure a render target partial resolve pass before binding a cleared RT to texture.
	Exists If:	[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'										
	Format:	S31 (2's complement) for all SINT surface formats										
	Format:	U32 for all UINT surface formats										
Format:	IEEE_FP for all other surface formats											
Programming Notes												
If programmed to non 0/1 values, SW must ensure a render target partial resolve pass before binding a cleared RT to texture.												

RENDER_SURFACE_STATE																	
	<table border="1"> <tr> <td style="text-align: center;">31:0</td> <td>Reserved</td> </tr> <tr> <td>Exists If:</td> <td>[Auxiliary Surface Mode] == 'AUX_HIZ'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	31:0	Reserved	Exists If:	[Auxiliary Surface Mode] == 'AUX_HIZ'	Format:	MBZ										
31:0	Reserved																
Exists If:	[Auxiliary Surface Mode] == 'AUX_HIZ'																
Format:	MBZ																
14	<table border="1"> <tr> <td style="text-align: center;">31:0</td> <td>Blue Clear Color</td> </tr> <tr> <td>Exists If:</td> <td>[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'</td> </tr> <tr> <td>Format:</td> <td>S31 (2's complement) for all SINT surface formats</td> </tr> <tr> <td>Format:</td> <td>U32 for all UINT surface formats</td> </tr> <tr> <td>Format:</td> <td>IEEE_FP for all other surface formats</td> </tr> <tr> <td colspan="2"> For Sampling Engine Surfaces and Render Targets with Auxiliary Surface Mode set to AUX_CCS: Specifies the clear value for the blue channel. For Other Surfaces: This field is ignored. </td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">If programmed to non 0/1 values, SW must ensure a render target partial resolve pass before binding a cleared RT to texture.</td> </tr> </table>	31:0	Blue Clear Color	Exists If:	[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'	Format:	S31 (2's complement) for all SINT surface formats	Format:	U32 for all UINT surface formats	Format:	IEEE_FP for all other surface formats	For Sampling Engine Surfaces and Render Targets with Auxiliary Surface Mode set to AUX_CCS: Specifies the clear value for the blue channel. For Other Surfaces: This field is ignored.		Programming Notes		If programmed to non 0/1 values, SW must ensure a render target partial resolve pass before binding a cleared RT to texture.	
	31:0	Blue Clear Color															
Exists If:	[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'																
Format:	S31 (2's complement) for all SINT surface formats																
Format:	U32 for all UINT surface formats																
Format:	IEEE_FP for all other surface formats																
For Sampling Engine Surfaces and Render Targets with Auxiliary Surface Mode set to AUX_CCS: Specifies the clear value for the blue channel. For Other Surfaces: This field is ignored.																	
Programming Notes																	
If programmed to non 0/1 values, SW must ensure a render target partial resolve pass before binding a cleared RT to texture.																	
	<table border="1"> <tr> <td style="text-align: center;">31:0</td> <td>Reserved</td> </tr> <tr> <td>Exists If:</td> <td>[Auxiliary Surface Mode] == 'AUX_HIZ'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	31:0	Reserved	Exists If:	[Auxiliary Surface Mode] == 'AUX_HIZ'	Format:	MBZ										
31:0	Reserved																
Exists If:	[Auxiliary Surface Mode] == 'AUX_HIZ'																
Format:	MBZ																
15	<table border="1"> <tr> <td style="text-align: center;">31:0</td> <td>Alpha Clear Color</td> </tr> <tr> <td>Exists If:</td> <td>[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'</td> </tr> <tr> <td>Format:</td> <td>S31 (2's complement) for all SINT surface formats</td> </tr> <tr> <td>Format:</td> <td>U32 for all UINT surface formats</td> </tr> <tr> <td>Format:</td> <td>IEEE_FP for all other surface formats</td> </tr> <tr> <td colspan="2"> For Sampling Engine Surfaces and Render Targets with Auxiliary Surface Mode set to AUX_CCS: Specifies the clear value for the alpha channel. For Other Surfaces: This field is ignored. </td> </tr> <tr> <td colspan="2" style="text-align: center;">Programming Notes</td> </tr> <tr> <td colspan="2">If programmed to non 0/1 values, SW must ensure a render target partial resolve pass before binding a cleared RT to texture.</td> </tr> </table>	31:0	Alpha Clear Color	Exists If:	[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'	Format:	S31 (2's complement) for all SINT surface formats	Format:	U32 for all UINT surface formats	Format:	IEEE_FP for all other surface formats	For Sampling Engine Surfaces and Render Targets with Auxiliary Surface Mode set to AUX_CCS: Specifies the clear value for the alpha channel. For Other Surfaces: This field is ignored.		Programming Notes		If programmed to non 0/1 values, SW must ensure a render target partial resolve pass before binding a cleared RT to texture.	
	31:0	Alpha Clear Color															
Exists If:	[Auxiliary Surface Mode] == 'AUX_CCS_D' OR [Auxiliary Surface Mode] == 'AUX_CCS_E'																
Format:	S31 (2's complement) for all SINT surface formats																
Format:	U32 for all UINT surface formats																
Format:	IEEE_FP for all other surface formats																
For Sampling Engine Surfaces and Render Targets with Auxiliary Surface Mode set to AUX_CCS: Specifies the clear value for the alpha channel. For Other Surfaces: This field is ignored.																	
Programming Notes																	
If programmed to non 0/1 values, SW must ensure a render target partial resolve pass before binding a cleared RT to texture.																	
	<table border="1"> <tr> <td style="text-align: center;">31:0</td> <td>Reserved</td> </tr> <tr> <td>Exists If:</td> <td>[Auxiliary Surface Mode] == 'AUX_HIZ'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	31:0	Reserved	Exists If:	[Auxiliary Surface Mode] == 'AUX_HIZ'	Format:	MBZ										
31:0	Reserved																
Exists If:	[Auxiliary Surface Mode] == 'AUX_HIZ'																
Format:	MBZ																

Render Data Port Message Types

MT_DP_RT - Render Data Port Message Types															
Source:	Render Cache DataPort														
Size (in bits):	5														
Default Value:	0x0000000C														
Lists all the Message Types in a Render Data Port Message Descriptor [18:14].															
DWord	Bit	Description													
0	4	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ											
	Format:	MBZ													
3:0	<p>Message Type</p> <table border="1"> <tr> <td>Format:</td> <td>Enumeration</td> </tr> </table> <p>Specifies type of message</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0Ch</td> <td>MT_RTW [Default]</td> <td>Render Target Write message</td> </tr> <tr> <td>0Dh</td> <td>MT_RTR</td> <td>Render Target Read message</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Format:	Enumeration	Value	Name	Description	0Ch	MT_RTW [Default]	Render Target Write message	0Dh	MT_RTR	Render Target Read message	Others	Reserved	Ignored
Format:	Enumeration														
Value	Name	Description													
0Ch	MT_RTW [Default]	Render Target Write message													
0Dh	MT_RTR	Render Target Read message													
Others	Reserved	Ignored													

Render Target Index Message Header Control

MHC_RT_RTI - Render Target Index Message Header Control		
Source:	BSpec	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:3	Reserved
		<table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>
Format:	Ignore	
	2:0	Render Target Index
		<table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Specifies the render target index that will be used to select blend state from BLEND_STATE.</p>
Format:	U3	

Render Target Message Header

MH_RT - Render Target Message Header		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0	31:0	Render Target Controls 0 Format: MHC_RT_CO Specifies controls for Render Target Write and Read messages.
0.1	31:0	Color Calculator State Pointer Format: MHC_RT_CCSP For Render Target Write message, specifies the HWORD-aligned GeneralStateOffset for Color State. Ignored by Render Target Read message.
0.2	31:0	Render Target Index Format: MHC_RT_RTI For Render Target Write message, specifies the render target index used to select blend state from BLEND_STATE. Ignored by Render Target Read message.
0.3-0.4	63:0	Reserved Format: ignore Ignored
0.5	31:0	Color Code Format: MHC_RT_CC Hardware uses to track synchronizing events and free resources on thread completion.
0.6-0.7	63:0	Reserved Format: ignore Ignored
1.0-1.1	63:0	Reserved Format: ignore Ignored
1.2	31:0	Subspan 0 Format: MHC_RT_SUBSPAN Upper left corner of subspan 0
1.3	31:0	Subspan 1 Format: MHC_RT_SUBSPAN Upper left corner of subspan 1

MH_RT - Render Target Message Header		
1.4	31:0	Subspan 2
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MHC_RT_SUBSPAN</td> </tr> </table> <p>Upper left corner of subspan 2</p>
Format:	MHC_RT_SUBSPAN	
1.5	31:0	Subspan 3
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MHC_RT_SUBSPAN</td> </tr> </table> <p>Upper left corner of subspan 3</p>
Format:	MHC_RT_SUBSPAN	
1.6	31:0	Reserved
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">Ignore</td> </tr> </table> <p>Ignored</p>
Format:	Ignore	
1.7	31:0	Pixel Sample Enables
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MHC_RT_PSM</td> </tr> </table> <p>Pixel Sample Enables</p>
Format:	MHC_RT_PSM	

Render Target Message Header Control

MHC_RT_C0 - Render Target Message Header Control												
Source:	BSpec											
Size (in bits):	32											
Default Value:	0x00000000											
DWord	Bit	Description										
0	31	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore								
	Format:	Ignore										
	30:27	<p>Viewport Index</p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>For Render Target Write message, specifies the index of the viewport currently being used. Range = [0,15] Ignored by Render Target Read message.</p>	Format:	U4								
	Format:	U4										
	26:16	<p>Render Target Array Index</p> <table border="1"> <tr> <td>Format:</td> <td>U11</td> </tr> </table> <p>Specifies the array index to be used for the following surface types: SURFTYPE_1D: specifies the array index. Range = [0,511] SURFTYPE_2D: specifies the array index. Range = [0,511] SURFTYPE_3D: specifies the Z or R coordinate. Range = [0,2047] SURFTYPE_BUFFER: must be zero. SURFTYPE_CUBE: specifies the face identifier. Mapping (0,+x) (1,-x) (2,+y) (3,-y) (4,+z) (5,-z).</p> <table border="1"> <thead> <tr> <th colspan="3">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="3">The Render Target Array Index used by hardware for access to the Render Target is overridden with the Minimum Array Element defined in SURFACE_STATE if it is out of the range between Minimum Array Element and Depth. For cube surfaces, a depth value of 5 is used for this determination.</td> </tr> </tbody> </table>	Format:	U11	Programming Notes			The Render Target Array Index used by hardware for access to the Render Target is overridden with the Minimum Array Element defined in SURFACE_STATE if it is out of the range between Minimum Array Element and Depth. For cube surfaces, a depth value of 5 is used for this determination.				
Format:	U11											
Programming Notes												
The Render Target Array Index used by hardware for access to the Render Target is overridden with the Minimum Array Element defined in SURFACE_STATE if it is out of the range between Minimum Array Element and Depth. For cube surfaces, a depth value of 5 is used for this determination.												
15	<p>Front/Back Facing Polygon</p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Determines whether the polygon is front or back facing. Used by the render cache to determine which stencil test state to use.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Front facing</td> <td>All</td> </tr> <tr> <td>1h</td> <td>Back facing</td> <td>All</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	0h	Front facing	All	1h	Back facing	All
Format:	U1											
Value	Name	Description										
0h	Front facing	All										
1h	Back facing	All										
14	<p>Stencil Present to Render Target</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For Render Target Write message, indicates that computed stencil is included in the message. Must be zero for Render Target Read message.</p>	Format:	Enable									
Format:	Enable											
13	<p>Source Depth Present to Render Target</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>For Render Target Write Message, indicates that source depth data is included in the message. Must be zero for Render Target Read message.</p>	Format:	Enable									
Format:	Enable											

MHC_RT_C0 - Render Target Message Header Control				
12	<p>oMask to Render Target</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>For Render Target Write message, indicates that oMask data is present in the message and is to be used to mask off samples. Must be zero for Render Target Read message.</p>	Format:	Enable	
Format:	Enable			
11	<p>Source0 Alpha Present to Render Target</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>For Render Target Write message, indicates that Source0 Alpha (aka o0.a) data is included in RTWrite message. If present, these alpha values are used as inputs to AlphaTest and AlphaToCoverage functions. This is required to meet the API rules when writing to multiple render targets (MRTs). Must be zero for Render Target Read message.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This bit should not be set when write to RT0, though sending and using redundant alpha will provide the correct results (at lower performance). This bit is not supported on Dual-Source Blend message types, as source0 alpha is already included in those messages. This bit is not supported on replicated data message types.</p>	Format:	Enable	Programming Notes
Format:	Enable			
Programming Notes				
10	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore	
Format:	Ignore			
9:6	<p>Starting Sample Pair Index or Sample Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U4</td> </tr> </table> <p>When pixel shader is dispatched in per-sample mode or per-pixel mode with Per-Sample PS Enable bit cleared, this field indicates the index of the first sample pair of the dispatch. Range = [0,7].</p> <p>When pixel shader is dispatched in per-pixel mode with Per-Sample PS Enable bit set, this field indicates the index of a sample referenced by per-sample RT read or RT write messages. Range = [0, 15].</p>	Format:	U4	
Format:	U4			
5:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore	
Format:	Ignore			

Replicated Pixel Render Target Data Payload Register

MDPR_RGBA - Replicated Pixel Render Target Data Payload Register				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	<p>Red</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U32</td> </tr> </table> <p>Specifies the value of all slots' red channel.</p>	Format:	U32
Format:	U32			
1	31:0	<p>Green</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U32</td> </tr> </table> <p>Specifies the value of all slots' green channel.</p>	Format:	U32
Format:	U32			
2	31:0	<p>Blue</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U32</td> </tr> </table> <p>Specifies the value of all slots' blue channel.</p>	Format:	U32
Format:	U32			
3	31:0	<p>Alpha</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U32</td> </tr> </table> <p>Specifies the value of all slots' alpha channel.</p>	Format:	U32
Format:	U32			
4-7	127:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

Replicated SIMD16 Render Target Data Payload

MDP_RTW_16REP - Replicated SIMD16 Render Target Data Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	RGBA <table border="1" data-bbox="657 699 1469 747"> <tr> <td>Format:</td> <td>MDPR_RGBA</td> </tr> </table> RGBA for all slots [15:0]	Format:	MDPR_RGBA
Format:	MDPR_RGBA			

Reversed SIMD Mode 2 Message Descriptor Control Field

MDC_SM2R - Reversed SIMD Mode 2 Message Descriptor Control Field											
Source:	BSpec										
Size (in bits):	1										
Default Value:	0x00000000										
DWord	Bit	Description									
0	0	SIMD Mode									
		Format: Enumeration									
		Specifies the SIMD mode of the message (number of slots processed)									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>SIMD16</td> <td>SIMD16</td> </tr> <tr> <td>01h</td> <td>SIMD8</td> <td>SIMD8</td> </tr> </tbody> </table>	Value	Name	Description	00h	SIMD16	SIMD16	01h	SIMD8	SIMD8
		Value	Name	Description							
00h	SIMD16	SIMD16									
01h	SIMD8	SIMD8									

RoundingPrecisionTable_3_Bits

RoundingPrecisionTable_3_Bits																						
Source:	BSpec																					
Size (in bits):	3																					
Default Value:	0x00000000																					
DWord	Bit	Description																				
0	2:0	Rounding Precision <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U3</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>000b</td> <td>+1/16</td> </tr> <tr> <td>001b</td> <td>+2/16</td> </tr> <tr> <td>010b</td> <td>+3/16</td> </tr> <tr> <td>011b</td> <td>+4/16</td> </tr> <tr> <td>100b</td> <td>+5/16</td> </tr> <tr> <td>101b</td> <td>+6/16</td> </tr> <tr> <td>110b</td> <td>+7/16</td> </tr> <tr> <td>111b</td> <td>+8/16</td> </tr> </table>	Format:	U3	Value	Name	000b	+1/16	001b	+2/16	010b	+3/16	011b	+4/16	100b	+5/16	101b	+6/16	110b	+7/16	111b	+8/16
Format:	U3																					
Value	Name																					
000b	+1/16																					
001b	+2/16																					
010b	+3/16																					
011b	+4/16																					
100b	+5/16																					
101b	+6/16																					
110b	+7/16																					
111b	+8/16																					

MDP_RTW_A16 - S0A SIMD16 Render Target Data Payload				
7.0-7.7	255:0	Blue[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Blue	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
8.0-8.7	255:0	Alpha[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Alpha	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
9.0-9.7	255:0	Alpha[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Alpha	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			

SAMPLER_BORDER_COLOR_STATE

SAMPLER_BORDER_COLOR_STATE						
Source:	BSpec					
Size (in bits):	128					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000					
Description						
<p>The interpretation of the border color depends on the Texture Border Color Mode field in SAMPLER_STATE as follows:</p> <ul style="list-style-type: none"> • DX9 mode: The border color is 8-bit UNORM format, regardless of the surface format chosen. For surface formats with one or more channels missing (i.e. R5G6R5_UNORM is missing the alpha channel), the value from the border color, if selected, will be used even for the missing channels. • DX10/OpenGL mode: the format of the border color depends on the format of the surface being sampled. If the map format is UINT, then the border color format is R32G32B32A32_UINT. If the map format is SINT, then the border color format is R32G32B32A32_SINT. Otherwise, the border color format is R32G32B32A32_FLOAT. For surface formats with one or more channels missing, the value from the border color is not used for the missing channels, resulting in these channels resulting in the overall default value (0 for colors and 1 for alpha) regardless of whether border color is chosen. The surface formats with "L" and "I" have special behavior with respect to the border color. The border color value used for the replicated channels (RGB for "L" formats and RGBA for "I" formats) comes from the red channel of border color. In these cases, the green and blue channels, and also alpha for "I", of the border color are ignored. The format of this state depends on the Texture Border Color Mode field. 						
Programming Notes						
<ul style="list-style-type: none"> • DX9 mode is not supported for surfaces with more than 16 bits in any channel, other than 32-bit float formats which are supported. • The conditions under which this color is used depend on the Surface Type - 1D/2D/3D surfaces use the border color when the coordinates extend beyond the surface extent; cube surfaces use the border color for "empty" (disabled) faces. • The border color itself is accessed through the texture cache hierarchy rather than the state cache hierarchy. Thus, if the border color is changed in memory, the texture cache must be invalidated and the state cache does not need to be invalidated. • MAPFILTER_MONO: The border color is ignored. Border color is fixed at a value of 0 by hardware. 						
DWord	Bit	Description				
0	31:24	<p>Border Color Alpha</p> <table border="1"> <tr> <td>Exists If:</td> <td>Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> <p>Texture Border Color Mode = DX9</p>	Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	UNORM8
Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'					
Format:	UNORM8					

SAMPLER_BORDER_COLOR_STATE						
	23:16	<p>Border Color Blue</p> <table border="1"> <tr> <td>Exists If:</td> <td>Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> <p>Texture Border Color Mode = DX9</p>	Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	UNORM8
	Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'				
	Format:	UNORM8				
	15:8	<p>Border Color Green</p> <table border="1"> <tr> <td>Exists If:</td> <td>Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> <p>Texture Border Color Mode = DX9</p>	Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	UNORM8
Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'					
Format:	UNORM8					
31:0	<p>Border Color Red - (DX10/OGL)</p> <table border="1"> <tr> <td>Exists If:</td> <td>Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL'</td> </tr> <tr> <td>Format:</td> <td>IEEE_FP</td> </tr> </table> <p>Texture Border Color Mode = DX10/OGL</p>	Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL'	Format:	IEEE_FP	
Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL'					
Format:	IEEE_FP					
7:0	<p>Border Color Red - (DX9)</p> <table border="1"> <tr> <td>Exists If:</td> <td>Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> <p>Texture Border Color Mode = DX9</p>	Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	UNORM8	
Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'					
Format:	UNORM8					
1	31:0	<p>Border Color Green</p> <table border="1"> <tr> <td>Format:</td> <td>IEEE_FP</td> </tr> </table> <p>Texture Border Color Mode = DX10/OGL</p>	Format:	IEEE_FP		
Format:	IEEE_FP					
2	31:0	<p>Border Color Blue</p> <table border="1"> <tr> <td>Format:</td> <td>IEEE_FP</td> </tr> </table> <p>Texture Border Color Mode = DX10/OGL</p>	Format:	IEEE_FP		
Format:	IEEE_FP					
3	31:0	<p>Border Color Alpha</p> <table border="1"> <tr> <td>Format:</td> <td>IEEE_FP</td> </tr> </table> <p>Texture Border Color Mode = DX10/OGL</p>	Format:	IEEE_FP		
Format:	IEEE_FP					

SAMPLER_INDIRECT_STATE_BORDER_COLOR

SAMPLER_INDIRECT_STATE_BORDER_COLOR		
Source:	BSpec	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
<p>This structure is a one version of the SAMPLER_INDIRECT_STATE structure, suitable for many needs. An instance of this structure is pointed to by the Indirect State Pointer field in SAMPLER_STATE. The interpretation of the border color depends on the Texture Border Color Mode field in SAMPLER_STATE as follows:</p> <ul style="list-style-type: none"> In DX9 mode, the border color is 8-bit UNORM format, regardless of the surface format chosen. For surface formats with one or more channels missing (i.e. R5G6R5_UNORM is missing the alpha channel), the value from the border color, if selected, will be used <i>even for the missing channels</i>. In DX10/OGL mode, the format of the border color is R32G32B32A32_FLOAT, R32G32B32A32_SINT, or R32G32B32A32_UINT, depending on the surface format chosen. For surface formats with one or more channels missing, the value from the border color is not used for the missing channels, resulting in these channels resulting in the overall default value (0 for colors and 1 for alpha) regardless of whether border color is chosen. The surface formats with "L" and "I" have special behavior with respect to the border color. The border color value used for the replicated channels (RGB for "L" formats and RGBA for "I" formats) comes from the <i>red</i> channel of border color. In these cases, the green and blue channels, and also alpha for "I", of the border color are ignored. 		
Programming Notes		
<ul style="list-style-type: none"> DX9 mode is not supported for surfaces with more than 16 bits in any channel, other than 32-bit float formats which are supported. The conditions under which this color is used depend on the Surface Type - 1D/2D/3D surfaces use the border color when the coordinates extend beyond the surface extent; cube surfaces use the border color for "empty" (disabled) faces. The border color itself is accessed through the texture cache hierarchy rather than the state cache hierarchy. Thus, if the border color is changed in memory, the texture cache must be invalidated and the state cache does not need to be invalidated. MAPFILTER_MONO: The border color is ignored. Border color is fixed at a value of 0 by hardware. 		
DWord	Bit	Description
0	31:24	Border Color Alpha As U8
		Exists If: //Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'
	Format: U8	
	23:16	Border Color Blue As U8
Exists If: //Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'		
Format: U8		

SAMPLER_INDIRECT_STATE_BORDER_COLOR						
	15:8	Border Color Green As U8 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	U8
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'				
	Format:	U8				
	31:0	Border Color Red As Float <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]== 'true')</td> </tr> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]== 'true')	Format:	IEEE_Float
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]== 'true')				
Format:	IEEE_Float					
31:0	Border Color Red As U32 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]== 'true')</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]== 'true')	Format:	U32	
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]== 'true')					
Format:	U32					
31:0	Border Color Red As S31 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')	Format:	S31	
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')					
Format:	S31					
7:0	Border Color Red As U8 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	U8	
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'					
Format:	U8					
1	31:0	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	MBZ
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'				
	Format:	MBZ				
	31:0	Border Color Green As S31 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')	Format:	S31
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')					
Format:	S31					
31:0	Border Color Green As U32 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]== 'true')</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]== 'true')	Format:	U32	
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]== 'true')					
Format:	U32					
31:0	Border Color Green As Float <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]== 'true')</td> </tr> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]== 'true')	Format:	IEEE_Float	
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]== 'true')					
Format:	IEEE_Float					
2	31:0	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	MBZ
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'				
Format:	MBZ					

SAMPLER_INDIRECT_STATE_BORDER_COLOR						
	31:0	Border Color Blue As S31 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]==true'</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]==true'	Format:	S31
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]==true'				
	Format:	S31				
	31:0	Border Color Blue As U32 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]==true'</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]==true'	Format:	U32
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]==true'				
	Format:	U32				
	31:0	Border Color Blue As Float <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]==true'</td> </tr> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]==true'	Format:	IEEE_Float
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]==true'				
	Format:	IEEE_Float				
3	31:0	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	MBZ
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'				
	Format:	MBZ				
	31:0	Border Color Alpha As S31 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]==true'</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]==true'	Format:	S31
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]==true'					
Format:	S31					
31:0	Border Color Alpha As U32 <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]==true'</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]==true'	Format:	U32	
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsUnsigned]==true'					
Format:	U32					
31:0	Border Color Alpha As Float <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]==true'</td> </tr> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]==true'	Format:	IEEE_Float	
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsFloat]==true'					
Format:	IEEE_Float					

SAMPLER_INDIRECT_STATE

SAMPLER_INDIRECT_STATE		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
<p>Note: There are three variations of this structure, defined separately because their payloads have different lengths. Currently only SAMPLER_INDIRECT_STATE_BORDER_COLOR is fully defined.</p> <p>This structure is pointed to by Indirect State Pointer (SAMPLER_STATE).</p> <p>The interpretation of the border color depends on the Texture Border Color Mode field in SAMPLER_STATE as follows:</p> <ul style="list-style-type: none"> In DX9 mode, the border color is 8-bit UNORM format, regardless of the surface format chosen. For surface formats with one or more channels missing (i.e. R5G6R5_UNORM is missing the alpha channel), the value from the border color, if selected, will be used <i>even for the missing channels</i>. In DX10/OGL mode, the format of the border color is R32G32B32A32_FLOAT, R32G32B32A32_SINT, or R32G32B32A32_UINT, depending on the surface format chosen. For surface formats with one or more channels missing, the value from the border color is not used for the missing channels, resulting in these channels resulting in the overall default value (0 for colors and 1 for alpha) regardless of whether border color is chosen. The surface formats with "L" and "I" have special behavior with respect to the border color. The border color value used for the replicated channels (RGB for "L" formats and RGBA for "I" formats) comes from the <i>red</i> channel of border color. In these cases, the green and blue channels, and also alpha for "I", of the border color are ignored. <p>The format of this state depends on the Texture Border Color Mode field.</p>		
Programming Notes		
<ul style="list-style-type: none"> DX9 mode is not supported for surfaces with more than 16 bits in any channel, other than 32-bit float formats which are supported. The conditions under which this color is used depend on the Surface Type - 1D/2D/3D surfaces use the border color when the coordinates extend beyond the surface extent; cube surfaces use the border color for "empty" (disabled) faces. The border color itself is accessed through the texture cache hierarchy rather than the state cache hierarchy. Thus, if the border color is changed in memory, the texture cache must be invalidated and the state cache does not need to be invalidated. MAPFILTER_MONO: The border color is ignored. Border color is fixed at a value of 0 by hardware. 		
DWord	Bit	Description
0	31:24	Border Color Alpha
		Exists If: //Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'
		Format: UNORM8
Texture Border Color Mode = DX9		

SAMPLER_INDIRECT_STATE										
	23:16	Border Color Blue <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> Texture Border Color Mode = DX9	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	UNORM8				
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'								
	Format:	UNORM8								
	15:8	Border Color Green <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> Texture Border Color Mode = DX9	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	UNORM8				
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'									
Format:	UNORM8									
31:0	Border Color Red <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'</td> </tr> <tr> <td>Format:</td> <td>SINT32 (2's complement) for all SINT surface formats</td> </tr> <tr> <td>Format:</td> <td>UINT32 for all UINT surface formats</td> </tr> <tr> <td>Format:</td> <td>IEEE_FP for all other surface formats</td> </tr> </table> Texture Border Color Mode = DX10/OpenGL	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'	Format:	SINT32 (2's complement) for all SINT surface formats	Format:	UINT32 for all UINT surface formats	Format:	IEEE_FP for all other surface formats	
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'									
Format:	SINT32 (2's complement) for all SINT surface formats									
Format:	UINT32 for all UINT surface formats									
Format:	IEEE_FP for all other surface formats									
	7:0	Border Color Red <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> Texture Border Color Mode = DX9	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	UNORM8				
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'									
Format:	UNORM8									
1	31:0	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	MBZ				
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'								
Format:	MBZ									
	31:0	Border Color Green <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'</td> </tr> <tr> <td>Format:</td> <td>IEEE_FP</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table> Texture Border Color Mode = DX10/OpenGL	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'	Format:	IEEE_FP	Format:	S31	Format:	U32
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'									
Format:	IEEE_FP									
Format:	S31									
Format:	U32									
2	31:0	Reserved <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	MBZ				
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'								
Format:	MBZ									
	31:0	Border Color Blue <table border="1"> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'</td> </tr> <tr> <td>Format:</td> <td>IEEE_FP</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table> Texture Border Color Mode = DX10/OpenGL	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'	Format:	IEEE_FP	Format:	S31	Format:	U32
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'									
Format:	IEEE_FP									
Format:	S31									
Format:	U32									

SAMPLER_INDIRECT_STATE									
3	31:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'	Format:	MBZ			
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX9'							
Format:	MBZ								
31:0	<p>Border Color Alpha</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'</td> </tr> <tr> <td>Format:</td> <td>IEEE_FP</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Texture Border Color Mode = DX10/OpenGL</p>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'	Format:	IEEE_FP	Format:	S31	Format:	U32
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OpenGL'								
Format:	IEEE_FP								
Format:	S31								
Format:	U32								
4..15	31:0	Reserved							

SAMPLER_STATE_8x8_AVS_COEFFICIENTS

SAMPLER_STATE_8x8_AVS_COEFFICIENTS		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
Description		
ExistsIf = AVS && (Function_mode = 0)		
DWord	Bit	Description
0	31:24	Table 0Y Filter Coefficient[n,1]
		Format: S1.6 2's Complement Range: [-2, +2)
	23:16	Table 0X Filter Coefficient[n,1]
		Format: S1.6 2's Complement Range: [-2, +2)
15:8	Table 0Y Filter Coefficient[n,0]	
	Format: S1.6 2's Complement Range: [-2, +2) Programming Notes If the format is R10G10B10A2_UNORM or R8G8B8A8_UNORM, this field MBZ.	
7:0	Table 0X Filter Coefficient[n,0]	
	Format: S1.6 2's Complement Range: [-2, +2) Programming Notes If the format is R10G10B10A2_UNORM or R8G8B8A8_UNORM, this field MBZ.	
1	31:24	Table 0Y Filter Coefficient[n,3]
		Format: S1.6 2's Complement Range: [-2.0, +2.0)
	23:16	Table 0X Filter Coefficient[n,3]
Format: S1.6 2's Complement Range: [-2.0, +2.0)		
15:8	Table 0Y Filter Coefficient[n,2]	
		Format: S1.6 2's Complement Range: [-2.0, +2.0)

SAMPLER_STATE_8x8_AVS_COEFFICIENTS		
	7:0	Table 0X Filter Coefficient[n,2]
		Format: S1.6 2's Complement Range: [-2.0, +2.0)
2	31:24	Table 0Y Filter Coefficient[n,5]
		Format: S1.6 2's Complement Range: [-2.0, +2.0)
	23:16	Table 0X Filter Coefficient[n,5]
		Format: S1.6 2's Complement Range: [-2.0, +2.0)
	15:8	Table 0Y Filter Coefficient[n,4]
		Format: S1.6 2's Complement Range: [-2.0, +2.0)
		Programming Notes
7:0	Table 0X Filter Coefficient[n,4]	
	Format: S1.6 2's Complement Range: [-2.0, +2.0)	
		Programming Notes
		If the format is R10G10B10A2_UNORM or R8G8B8A8_UNORM, this field MBZ.
3	31:24	Table 0Y Filter Coefficient[n,7]
		Format: S1.6 2's Complement Range: [-2, +2)
	23:16	Table 0X Filter Coefficient[n,7]
		Format: S1.6 2's Complement Range: [-2, +2)
15:8	Table 0Y Filter Coefficient[n,6]	
		Format: S1.6 2's Complement Range: [-2, +2)
7:0	Table 0X Filter Coefficient[n,6]	
		Format: S1.6 2's Complement Range: [-2, +2)

SAMPLER_STATE_8x8_AVS_COEFFICIENTS		
4	31:24	Table 1X Filter Coefficient[n,3] Format: S1.6 2's Complement Range: [-2.0, +2.0)
	23:16	Table 1X Filter Coefficient[n,2] Format: S1.6 2's Complement <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Description</div> Range: [-2.0, +2.0)
	15:0	Reserved Format: MBZ
5	31:16	Reserved Format: MBZ
	15:8	Table 1X Filter Coefficient[n,5] Format: S1.6 2's Complement <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Description</div> Range: [-2.0, +2.0)
	7:0	Table 1X Filter Coefficient[n,4] Format: S1.6 2's Complement Range: [-2.0, +2.0)
6	31:24	Table 1Y Filter Coefficient[n,3] Format: S1.6 2's Complement Range: [-2.0, +2.0)
	23:16	Table 1Y Filter Coefficient[n,2] Format: S1.6 2's Complement <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Description</div> Range: [-2.0, +2.0)
	15:0	Reserved Format: MBZ
7	31:16	Reserved Format: MBZ

SAMPLER_STATE_8x8_AVS_COEFFICIENTS		
	15:8	Table 1Y Filter Coefficient[n,5]
		Format: S1.6 2's Complement
		Description
		Range: [-2.0, +2.0)
	7:0	Table 1Y Filter Coefficient[n,4]
		Format: S1.6 2's Complement
		Range: [-2.0, +2.0)

SAMPLER_STATE_8x8_AVS			
0x00000000, 0x00000000			
Description			
ExistsIf = AVS && (Function_mode = 0)			
DWord	Bit	Description	
0	31:28	Reserved Format: MBZ	
	27:23	R3c Coefficient Default Value: 5 Format: U0.5 IEF smoothing coefficient, see IEF map.	
		22:18	R3x Coefficient Default Value: 5 Format: U0.5 IEF smoothing coefficient, see IEF map.
			17:12
	11:6	Weak Edge Threshold Default Value: 1 Format: U6 If Strong Edge Threshold > EM > Weak Edge Threshold , the basic VSA detects a weak edge.	
		5:0	Gain Factor Default Value: 44 Format: U6 User control sharpening strength
1	31:0	Reserved Format: MBZ	

SAMPLER_STATE_8x8_AVS		
2	31:27	R5c Coefficient
		Default Value: 7 Format: U0.5 IEF smoothing coefficient, see IEF map.
	26:22	R5cx Coefficient
		Default Value: 7 Format: U0.5 IEF smoothing coefficient, see IEF map.
	21:17	R5x Coefficient
		Default Value: 7 Format: U0.5 IEF smoothing coefficient, see IEF map.
	16:14	Strong Edge Weight
Default Value: 7 Format: U3 Sharpening strength when a strong edge is found in basic VSA.		
13:11	Regular Weight	
	Default Value: 2 Format: U3 Sharpening strength when a weak edge is found in basic VSA.	
10:8	Non Edge Weight	
	Default Value: 1 Format: U3 Sharpening strength when no edge is found in basic VSA.	
7:0	Global Noise Estimation	
	Default Value: 255 Format: U8 Global noise estimation of previous frame.	
3	31	Skin Tone Tuned IEF _ Enable
		Default Value: 1 Format: U1 Control bit to enable the skin tone tuned IEF.

SAMPLER_STATE_8x8_AVS				
30	IEF4Smooth_Enable			
	Format:	U1		
	Value	Name	Description	
	0	[Default]	IEF is operating as a content adaptive detail filter based on 5x5 region	
	1		IEF is operating as a content adaptive smooth filter based on 3x3 region	
	29:28	Enable 8-tap filter		
		Adaptive Filtering (Mode = 11) ExistsIf: R10G10B10A2_UNORM R8G8B8A8_UNORM (AYUV also) R8B8G8A8_UNORM B8G8R8A8_UNORM R16G16B16A16		
		Enable 8-tap Filtering on UV channel (Mode = 10) ExistsIf: R10G10B10A2_UNORM R8G8B8A8_UNORM (AYUV also) R8B8_UNORM (CrCb) R8_UNORM R8B8G8A8_UNORM B8G8R8A8_UNORM R16G16B16A16 Y8_UNORM		
		Value	Name	Description
		00b		4-tap filter is only done on all channels.
01b			Enable 8-tap Adaptive filter on G-channel. 4-tap filter on other channels.	
10b			8-tap filter is done on all channels (UV-ch uses the Y-coefficients)	
11b			Enable 8-tap Adaptive filter all channels (UV-ch uses the Y-coefficients).	
Programming Notes				
For 00 and 10, are applicable for RGB surfaces only or surface without Y-ch. In case it is a YUV surface it will default to adaptive mode automatically which is 01 and 11 respectively. Alpha channel is always bi-linear filter irrespective of the above modes.				
Mode 01 and 00 are legacy support and are supported on all surface formats.				
When Mode is 10 and Surface format is Y8_UNORM, Bypass X/Y Adaptive Filtering must be 1, and Default Sharp Level must be 255				
27:22	Hue_Max			
	Default Value:	14		
	Format:	U6		
Rectangle half width.				
21:16	Sat_Max			
	Default Value:	31		
	Format:	U6		
Rectangle half length				
15:8	Cos(alpha)			
	Format:	S0.7 2's Complement		
Default Value: 79/128				

SAMPLER_STATE_8x8_AV5													
	7:0	<p>Sin(alpha)</p> <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										
	23:16	<p>U_Mid</p> <table border="1"> <tr> <td>Default Value:</td> <td>110</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Rectangle middle-point U coordinate.</p>	Default Value:	110	Format:	U8							
		Default Value:	110										
	Format:	U8											
15	<p>VY_STD_Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Enables STD in the VY subspace.</p>	Format:	Enable										
Format:	Enable												
14:12	<p>Diamond Margin</p> <table border="1"> <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	<p>Shuffle_OutputWriteback for sample_8x8</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>Writeback same as Original Sample_8x8</td> </tr> <tr> <td>1</td> <td></td> <td>Writeback of Sample_8x8 Is Modified to Suite Sample_Unorm</td> </tr> </tbody> </table>		Value	Name	Description	0		Writeback same as Original Sample_8x8	1		Writeback of Sample_8x8 Is Modified to Suite Sample_Unorm		
	Value	Name	Description										
	0		Writeback same as Original Sample_8x8										
	1		Writeback of Sample_8x8 Is Modified to Suite Sample_Unorm										
10:0	<p>S3U</p> <table border="1"> <tr> <td>Format:</td> <td>S2.8 2's Complement</td> </tr> </table> <p>Default Value: 0/256</p>	Format:	S2.8 2's Complement										
Format:	S2.8 2's Complement												
5	31	<p>SkinDetailFactor</p> <table border="1"> <tr> <td>Format:</td> <td>S0</td> </tr> </table> <p>This flag bit is in operation only when the control bit Skin Tone TunedIEF_Enable is on.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> <td>sign(SkinDetailFactor) is equal to +1, and the content of the detected skin tone area is not detail revealed.</td> </tr> <tr> <td>0</td> <td></td> <td>sign(SkinDetailFactor) is equal to -1, and the content of the detected skin tone area is detail revealed.</td> </tr> </tbody> </table>	Format:	S0	Value	Name	Description	1		sign(SkinDetailFactor) is equal to +1, and the content of the detected skin tone area is not detail revealed.	0		sign(SkinDetailFactor) is equal to -1, and the content of the detected skin tone area is detail revealed.
		Format:	S0										
		Value	Name	Description									
		1		sign(SkinDetailFactor) is equal to +1, and the content of the detected skin tone area is not detail revealed.									
0		sign(SkinDetailFactor) is equal to -1, and the content of the detected skin tone area is detail revealed.											

SAMPLER_STATE_8x8_AVS					
	<p>30:24 Diamond_du</p> <table border="1"> <tr> <td>Default Value:</td> <td>2</td> </tr> <tr> <td>Format:</td> <td>S6 2's Complement</td> </tr> </table> <p>Rhombus center shift in the sat-direction, relative to the rectangle center.</p>	Default Value:	2	Format:	S6 2's Complement
	Default Value:	2			
	Format:	S6 2's Complement			
	<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>Default Value: 100/64 1 / tan(β)</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				

SAMPLER_STATE_8x8_AVS		
	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 $1/\text{Margin_VYL} = 3300/65536$
8	31:24	P1L Default Value: 216 Format: U8 Y Point 1 of the lower part of the detection PWLF.
		P0L Default Value: 46 Format: U8 Y Point 0 of the lower part of the detection PWLF.
	15:0	INV_Margin_VYU $1/\text{Margin_VYU} = 1600/65536$
	31:24	B1L Default Value: 130 Format: U8 V Bias 1 of the lower part of the detection PWLF.
		B0L Default Value: 133 Format: U8 V Bias 0 of the lower part of the detection PWLF.
		P3L Default Value: 236 Format: U8 Y Point 3 of the lower part of the detection PWLF.
9	15:8	P3L Default Value: 236 Format: U8 Y Point 3 of the lower part of the detection PWLF.

SAMPLER_STATE_8x8_AVS		
	7:0	P2L
		Default Value: 236
		Format: U8
		Y Point 2 of the lower part of the detection PWLF.
10	31:27	Y_Slope_2
		Format: U2.3
		Default Value: 31/8
		Slope between points Y3 and Y4.
	26:16	S0L
		Format: S2.8 2's Complement
		Default Value: -5/256
		Slope 0 of the lower part of the detection PWLF.
	15:8	B3L
		Default Value: 130
		Format: U8
		V Bias 3 of the lower part of the detection PWLF.
7:0	B2L	
	Default Value: 130	
	Format: U8	
11	31:22	Reserved
		Format: MBZ
	21:11	S2L
		Format: S2.8 2's Complement
		Default Value: 0/256
		Slope 2 of the lower part of the detection PWLF.
	10:0	S1L
		Format: S2.8 2's Complement
		Default Value: 0/256
Slope 1 of the lower part of the detection PWLF.		

SAMPLER_STATE_8x8_AVS				
12	31:27	Y_Slope1	Format: U2.3	
			Default Value: 31/8	
			Slope between points Y1 and Y2.	
	26:19	P1U	Default Value: 66	
		Format: U8		
		Y Point 1 of the upper part of the detection PWLF.		
18:11		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	
		Default Value: 0/256		
		Slope 3 of the lower part of the detection PWLF.		
13	31:24	B1U	Default Value: 163	
			Format: U8	
		V Bias 1 of the upper part of the detection PWLF.		
	23:16	B0U	Default Value: 143	
			Format: U8	
		V Bias 0 of the upper part of the detection PWLF.		
	15:8	P3U	Default Value: 236	
			Format: U8	
		Y Point 3 of the upper part of the detection PWLF.		
	7:0	P2U	Default Value: 150	
			Format: U8	
		Y Point 2 of the upper part of the detection PWLF.		

SAMPLER_STATE_8x8_AVS		
14	31:27	Reserved
		Format: MBZ
	26:16	S0U
		Format: S2.8 2's Complement
		Default Value: 256/256
		Slope 0 of the upper part of the detection PWLF.
	15:8	B3U
		Default Value: 140
		Format: U8 V Bias 3 of the upper part of the detection PWLF.
	7:0	B2U
		Default Value: 200
		Format: U8 V Bias 2 of the upper part of the detection PWLF.
15	31:22	Reserved
		Format: MBZ
	21:11	S2U
		Format: S2.8 2's Complement
		Default Value: -179/256 Slope 2 of the upper part of the detection PWLF.
	10:0	S1U
Format: S2.8 2's Complement		
Default Value: 113/256 Slope 1 of the upper part of the detection PWLF.		
16..23	255:0	Filter Coefficient[0]
		Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS
24..31	255:0	Filter Coefficient[1]
		Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS
32..39	255:0	Filter Coefficient[2]
		Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS
40..47	255:0	Filter Coefficient[3]
		Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS

SAMPLER_STATE_8x8_AVS											
48..55	255:0	Filter Coefficient[4] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
56..63	255:0	Filter Coefficient[5] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
64..71	255:0	Filter Coefficient[6] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
72..79	255:0	Filter Coefficient[7] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
80..87	255:0	Filter Coefficient[8] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
88..95	255:0	Filter Coefficient[9] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
96..103	255:0	Filter Coefficient[10] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
104..111	255:0	Filter Coefficient[11] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
112..119	255:0	Filter Coefficient[12] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
120..127	255:0	Filter Coefficient[13] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
128..135	255:0	Filter Coefficient[14] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
136..143	255:0	Filter Coefficient[15] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
144..151	255:0	Filter Coefficient[16] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS									
152	31:24	Default Sharpness Level Format: U8 When adaptive scaling is off, determines the balance between sharp and smooth scalars.									
		<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>Contribute 1 from the smooth scalar</td> </tr> <tr> <td>255</td> <td></td> <td>Contribute 1 from the sharp scalar</td> </tr> </tbody> </table>	Value	Name	Description	0	[Default]	Contribute 1 from the smooth scalar	255		Contribute 1 from the sharp scalar
		Value	Name	Description							
		0	[Default]	Contribute 1 from the smooth scalar							
		255		Contribute 1 from the sharp scalar							
Max Derivative 4 Pixels Format: U8 Used in adaptive filtering to specify the lower boundary of the smooth 4 pixel area.											

SAMPLER_STATE_8x8_AV5													
	15:8	<p>Max Derivative 8 Pixels</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Used in adaptive filtering to specify the lower boundary of the smooth 8 pixel area.</p>	Format:	U8									
	Format:	U8											
	7	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ											
	6:4	<p>Transition Area with 4 Pixels</p> <table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Used in adaptive filtering to specify the width of the transition area for the 4 pixel calculation.</p>	Format:	U3									
Format:	U3												
3	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												
2:0	<p>Transition Area with 8 Pixels</p> <table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Used in adaptive filtering to specify the width of the transition area for the 8 pixel calculation.</p>	Format:	U3										
Format:	U3												
153	31:23	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
	Format:	MBZ											
	22	<p>Bypass X Adaptive Filtering</p> <table border="1"> <tr> <td>Format:</td> <td>Disable</td> </tr> </table> <p>When disabled, the X direction will use Default Sharpness Level to blend between the smooth and sharp filters rather than the calculated value.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Disble</td> <td>Disable X Ddaptive Filtering</td> </tr> <tr> <td>0</td> <td>Enable</td> <td>Enable X Adaptive Filtering</td> </tr> </tbody> </table>	Format:	Disable	Value	Name	Description	1	Disble	Disable X Ddaptive Filtering	0	Enable	Enable X Adaptive Filtering
	Format:	Disable											
Value	Name	Description											
1	Disble	Disable X Ddaptive Filtering											
0	Enable	Enable X Adaptive Filtering											
21	<p>Bypass Y Adaptive Filtering</p> <table border="1"> <tr> <td>Format:</td> <td>Disable</td> </tr> </table> <p>When disabled, the Y direction will use Default Sharpness Level to blend between the smooth and sharp filters rather than the calculated value.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Disble</td> <td>Disable Y Ddaptive Filtering</td> </tr> <tr> <td>0</td> <td>Enable</td> <td>Enable Y Adaptive Filtering</td> </tr> </tbody> </table>	Format:	Disable	Value	Name	Description	1	Disble	Disable Y Ddaptive Filtering	0	Enable	Enable Y Adaptive Filtering	
Format:	Disable												
Value	Name	Description											
1	Disble	Disable Y Ddaptive Filtering											
0	Enable	Enable Y Adaptive Filtering											
20:2	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ										
Format:	MBZ												

SAMPLER_STATE_8x8_AVS											
	1	Adaptive Filter for all channels									
		Format: Enable									
		Only to be enabled if 8-tap Adaptive filter mode is on, eElse it should be disabled.									
		<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>1</td> <td>Enable</td> <td>Enable Adaptive Filter on UV/RB Channels</td> </tr> <tr> <td>0</td> <td>Disble</td> <td>Disable Adaptive Filter on UV/RB Channels</td> </tr> </tbody> </table>	Value	Name	Description	1	Enable	Enable Adaptive Filter on UV/RB Channels	0	Disble	Disable Adaptive Filter on UV/RB Channels
		Value	Name	Description							
1	Enable	Enable Adaptive Filter on UV/RB Channels									
0	Disble	Disable Adaptive Filter on UV/RB Channels									
0	RGB Adaptive										
Format: Enable											
This should be always set to 0 for YUV input and can be enabled/disabled for RGB input. This should be enabled only if we enable 8-tap adaptive filter for RGB input.											
<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>1</td> <td>Enable</td> <td>Enable the RGB Adaptive filter using the equation $(Y=(R+2G+B)\gg 2)$</td> </tr> <tr> <td>0</td> <td>Disble</td> <td>Disable the RGB Adaptive equation and use G-Ch directly for adaptive filter</td> </tr> </tbody> </table>		Value	Name	Description	1	Enable	Enable the RGB Adaptive filter using the equation $(Y=(R+2G+B)\gg 2)$	0	Disble	Disable the RGB Adaptive equation and use G-Ch directly for adaptive filter	
Value	Name	Description									
1	Enable	Enable the RGB Adaptive filter using the equation $(Y=(R+2G+B)\gg 2)$									
0	Disble	Disable the RGB Adaptive equation and use G-Ch directly for adaptive filter									
154..159	191:0	Reserved									
Format: MBZ											
160..167	255:0	Filter Coefficient[17]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											
168..175	255:0	Filter Coefficient[18]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											
176..183	255:0	Filter Coefficient[19]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											
184..191	255:0	Filter Coefficient[20]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											
192..199	255:0	Filter Coefficient[21]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											
200..207	255:0	Filter Coefficient[22]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											
208..215	255:0	Filter Coefficient[23]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											
216..223	255:0	Filter Coefficient[24]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											
224..231	255:0	Filter Coefficient[25]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											
232..239	255:0	Filter Coefficient[26]									
Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS											

SAMPLER_STATE_8x8_AVS		
240..247	255:0	Filter Coefficient[27] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS
248..255	255:0	Filter Coefficient[28] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS
256..263	255:0	Filter Coefficient[29] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS
264..271	255:0	Filter Coefficient[30] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS
272..279	255:0	Filter Coefficient[31] Format: SAMPLER_STATE_8x8_AVS_COEFFICIENTS

SAMPLER_STATE

SAMPLER_STATE																								
Source:	BSpec																							
Exists If:	//(MessageType != 'Deinterlace') && (MessageType != 'Sample_8x8')																							
Size (in bits):	128																							
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000																							
<p>This is the normal sampler state used by all messages that use SAMPLER_STATE except sample_8x8 and deinterlace. The sampler state is stored as an array of up to 16 elements, each of which contains the dwords described here. The start of each element is spaced 4 dwords apart. The first element of the sampler state array is aligned to a 32-byte boundary.</p>																								
DWord	Bit	Description																						
0	31	Sampler Disable <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Disable</td> </tr> </table> <p>This field allows the sampler to be disabled. If disabled, all output channels will return 0.</p>	Format:	Disable																				
	Format:	Disable																						
	30	Reserved																						
29	Texture Border Color Mode <p>For some surface formats, the 32 bit border color is decoded differently based on the border color mode. In addition, the default value of channels not included in the surface may be affected by this field. Refer to the "Sampler Output Channel Mapping" table for the values of these channels, and for surface formats that may only support one of these modes. Also refer to the definition of SAMPLER_BORDER_COLOR_STATE for more details on the behavior of the two modes defined by this field.</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>DX10/OGL</td> <td>DX10/OGL mode for interpreting the border color</td> </tr> <tr> <td>1h</td> <td>DX9</td> <td>DX9 and earlier mode for interpreting the border color</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 feild must not be set to DX9 if there are null tiles in use</td> </tr> <tr> <td colspan="2">This field is required to be the same for every message over a period of time. A flush of the sampler cache must occur before a message with the opposite state of this field is delivered.</td> </tr> <tr> <td colspan="2">This field must be set to DX9 mode when used with surfaces that have Surface Format P4A4_UNORM or A4P4_UNORM.</td> </tr> <tr> <td colspan="2">This field must be set to DX10/OGL mode when used with surfaces that have Surface Format YCRCB_SWAPUV or YCRCB_SWAPY.</td> </tr> <tr> <td colspan="2">This field must be set to DX10/OGL mode if Surface Format for the associated surface is UINT OR SINT.</td> </tr> <tr> <td colspan="2">This field must be set to DX10/OGL mode if REDUCTION_MINIMUM or REDUCTION_MAXIMUM or message type is sample_min or sample_max.</td> </tr> </tbody> </table>	Value	Name	Description	0h	DX10/OGL	DX10/OGL mode for interpreting the border color	1h	DX9	DX9 and earlier mode for interpreting the border color	Programming Notes		This feild must not be set to DX9 if there are null tiles in use		This field is required to be the same for every message over a period of time. A flush of the sampler cache must occur before a message with the opposite state of this field is delivered.		This field must be set to DX9 mode when used with surfaces that have Surface Format P4A4_UNORM or A4P4_UNORM.		This field must be set to DX10/OGL mode when used with surfaces that have Surface Format YCRCB_SWAPUV or YCRCB_SWAPY.		This field must be set to DX10/OGL mode if Surface Format for the associated surface is UINT OR SINT.		This field must be set to DX10/OGL mode if REDUCTION_MINIMUM or REDUCTION_MAXIMUM or message type is sample_min or sample_max.	
Value	Name	Description																						
0h	DX10/OGL	DX10/OGL mode for interpreting the border color																						
1h	DX9	DX9 and earlier mode for interpreting the border color																						
Programming Notes																								
This feild must not be set to DX9 if there are null tiles in use																								
This field is required to be the same for every message over a period of time. A flush of the sampler cache must occur before a message with the opposite state of this field is delivered.																								
This field must be set to DX9 mode when used with surfaces that have Surface Format P4A4_UNORM or A4P4_UNORM.																								
This field must be set to DX10/OGL mode when used with surfaces that have Surface Format YCRCB_SWAPUV or YCRCB_SWAPY.																								
This field must be set to DX10/OGL mode if Surface Format for the associated surface is UINT OR SINT.																								
This field must be set to DX10/OGL mode if REDUCTION_MINIMUM or REDUCTION_MAXIMUM or message type is sample_min or sample_max.																								

SAMPLER_STATE

28:27	<p>LOD PreClamp Mode</p> <p>This field determines whether the computed LOD is clamped to [max,min] mip level before the mag-vs-min determination is performed.</p> <p>PRECLAMP_OGL: LOD pre-clamped to Min LOD and Max LOD</p> <p>OpenGL API currently clamps LOD to the Min LOD and Max LOD (from Sampler State) prior to performing min/mag determination, and therefore it is expected that an OpenGL driver would need to set this field to PRECLAMP_OGL.</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>0h</td> <td>NONE</td> <td>LOD PreClamp disabled</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>2h</td> <td>OGL</td> <td>LOD PreClamp enabled (OGL mode)</td> </tr> </tbody> </table>	Value	Name	Description	0h	NONE	LOD PreClamp disabled	1h	Reserved		2h	OGL	LOD PreClamp enabled (OGL mode)					
Value	Name	Description																
0h	NONE	LOD PreClamp disabled																
1h	Reserved																	
2h	OGL	LOD PreClamp enabled (OGL mode)																
26:22	<p>Coarse LOD Quality Mode</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U5</td> </tr> </table> <p>This field configures the coarse LOD image quality mode for the sample_d, sample_l, and sample_b messages in the sampling engine. In general, performance will increase and power consumption will decrease with each step of reduced quality (performance gain for sample_l and sample_b will be minimal).</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 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>0h</td> <td>Disabled</td> <td>Full quality is enabled, matching prior products</td> </tr> <tr> <td>01h-1Fh</td> <td></td> <td>Quality degrades with each larger value, performance improves with each larger value</td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 5px; text-align: center; margin-bottom: 10px;">Programming Notes</div> <p>Although allowed, it is not recommended to program this field to a value greater than 17h to avoid masking the exponent which may generate incorrect LOD values.</p>	Format:	U5	Value	Name	Description	0h	Disabled	Full quality is enabled, matching prior products	01h-1Fh		Quality degrades with each larger value, performance improves with each larger value						
Format:	U5																	
Value	Name	Description																
0h	Disabled	Full quality is enabled, matching prior products																
01h-1Fh		Quality degrades with each larger value, performance improves with each larger value																
21:20	<p>Mip Mode Filter</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">U2 Enumerated Type</td> </tr> </table> <p>This field determines if and how mip map levels are chosen and/or combined when texture filtering.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <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>NONE</td> <td>Disable mip mapping - force use of the mipmap level corresponding to Min LOD.</td> </tr> <tr> <td>1h</td> <td>NEAREST</td> <td>Nearest, Select the nearest mip map</td> </tr> <tr> <td>2h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>3h</td> <td>LINEAR</td> <td>Linearly interpolate between nearest mip maps (combined with linear min/mag filters this is analogous to "Trilinear" filtering).</td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 5px; text-align: center; margin-bottom: 10px;">Programming Notes</div> <p>MIPFILTER_LINEAR is not supported for surface formats that do not support "Sampling Engine</p>	Format:	U2 Enumerated Type	Value	Name	Description	0h	NONE	Disable mip mapping - force use of the mipmap level corresponding to Min LOD.	1h	NEAREST	Nearest, Select the nearest mip map	2h	Reserved		3h	LINEAR	Linearly interpolate between nearest mip maps (combined with linear min/mag filters this is analogous to "Trilinear" filtering).
Format:	U2 Enumerated Type																	
Value	Name	Description																
0h	NONE	Disable mip mapping - force use of the mipmap level corresponding to Min LOD.																
1h	NEAREST	Nearest, Select the nearest mip map																
2h	Reserved																	
3h	LINEAR	Linearly interpolate between nearest mip maps (combined with linear min/mag filters this is analogous to "Trilinear" filtering).																

SAMPLER_STATE																								
	<p>Filtering" as indicated in the Surface Formats table unless using the sample_c message type or minimum/maximum operation.</p> <p>Mip Mode Filter must be set to MIPFILTER_NONE or MIPFILTER_NEAREST if Surface Format for the associated surface is UINT or SINT. However, all settings of this field are allowed with UINT/SINT if a minimum or maximum operation is being performed.</p> <p>Mip Mode Filter must be set to MIPFILTER_NONE for Planar YUV surfaces.</p>																							
19:17	<p>Mag Mode Filter</p> <table border="1"> <tr> <td>Format:</td> <td>U3 Enumerated Type</td> </tr> </table> <p>This field determines how texels are sampled/filtered when a texture is being "magnified" (enlarged). For volume maps, this filter mode selection also applies to the 3rd (inter-layer) dimension.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NEAREST</td> <td>Sample the nearest texel</td> </tr> <tr> <td>1h</td> <td>LINEAR</td> <td>Bilinearly filter the 4 nearest texels</td> </tr> <tr> <td>2h</td> <td>ANISOTROPIC</td> <td>Perform an "anisotropic" filter on the chosen mip level</td> </tr> <tr> <td>4h-5h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>6h</td> <td>MONO</td> <td>Perform a monochrome convolution filter</td> </tr> <tr> <td>7h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>Only MAPFILTER_NEAREST and MAPFILTER_LINEAR are supported for surfaces of type SURFTYPE_3D.</p> <p>Only MAPFILTER_NEAREST is supported for surface formats that do not support "Sampling Engine Filtering" as indicated in the Surface Formats table unless using the sample_c message type or minimum/maximum operation.</p> <p>MAPFILTER_MONO: Only CLAMP_BORDER texture addressing mode is supported. . Both Mag Mode Filter and Min Mode Filter must be programmed to MAPFILTER_MONO. Mip Mode Filter must be MIPFILTER_NONE. Only valid on surfaces with Surface Format MONO8 and with Surface Type SURFTYPE_2D.</p> <p>MAPFILTER_ANISOTROPIC may cause artifacts at cube edges if enabled for cube maps with the TEXCOORDMODE_CUBE addressing mode.</p> <p>MAPFILTER_ANISOTROPIC will be overridden to MAPFILTER_LINEAR when using a sample_l or sample_l_c message type or when Force LOD to Zero is set in the message header.</p> <p>Both Mag Mode Filter and Min Mode Filter must be set to MAPFILTER_NEAREST if Surface Format for the associated surface is UINT or SINT. However, all settings of this field other than MAPFILTER_MONO are allowed with UINT/SINT if a minimum or maximum operation is being performed.</p> <p>MAPFILTER_FLEXIBLE when float16 +/-inf apply to coefficient that are absolutely larger than 1.0 output result could be nan instead of +/-inf MAPFILTER_FLEXIBLE: A Null Tile reference will be reported back even if the associated texel has a coefficient of 0.0.</p>	Format:	U3 Enumerated Type	Value	Name	Description	0h	NEAREST	Sample the nearest texel	1h	LINEAR	Bilinearly filter the 4 nearest texels	2h	ANISOTROPIC	Perform an "anisotropic" filter on the chosen mip level	4h-5h	Reserved		6h	MONO	Perform a monochrome convolution filter	7h	Reserved	
Format:	U3 Enumerated Type																							
Value	Name	Description																						
0h	NEAREST	Sample the nearest texel																						
1h	LINEAR	Bilinearly filter the 4 nearest texels																						
2h	ANISOTROPIC	Perform an "anisotropic" filter on the chosen mip level																						
4h-5h	Reserved																							
6h	MONO	Perform a monochrome convolution filter																						
7h	Reserved																							

SAMPLER_STATE

	16:14	Min Mode Filter	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>U3 Enumerated Type</td> </tr> </table> <p>This field determines how texels are sampled/filtered when a texture is being "minified" (shrunk). For volume maps, this filter mode selection also applies to the 3rd (inter-layer) dimension. See Mag Mode Filter</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>NEAREST</td> <td>Sample the nearest texel</td> </tr> <tr> <td>1h</td> <td>LINEAR</td> <td>Bilinearly filter the 4 nearest texels</td> </tr> <tr> <td>2h</td> <td>ANISOTROPIC</td> <td>Perform an "anisotropic" filter on the chosen mip level</td> </tr> <tr> <td>4h-5h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>6h</td> <td>MONO</td> <td>Perform a monochrome convolution filter</td> </tr> <tr> <td>7h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <div style="text-align: center; background-color: #e6f2ff; padding: 5px; margin-top: 10px;">Programming Notes</div> <p>FLEXIBLE: A Null Tile reference will be reported back even if the associated texel has a coefficient of 0.0.</p>	Format:	U3 Enumerated Type	Value	Name	Description	0h	NEAREST	Sample the nearest texel	1h	LINEAR	Bilinearly filter the 4 nearest texels	2h	ANISOTROPIC	Perform an "anisotropic" filter on the chosen mip level	4h-5h	Reserved		6h	MONO	Perform a monochrome convolution filter	7h	Reserved	
Format:	U3 Enumerated Type																									
Value	Name	Description																								
0h	NEAREST	Sample the nearest texel																								
1h	LINEAR	Bilinearly filter the 4 nearest texels																								
2h	ANISOTROPIC	Perform an "anisotropic" filter on the chosen mip level																								
4h-5h	Reserved																									
6h	MONO	Perform a monochrome convolution filter																								
7h	Reserved																									
	13:1	Texture LOD Bias	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>S4.8 2's complement</td> </tr> </table> <p>Range: [-16.0, 16.0)</p> <p>This field specifies the signed bias value added to the calculated texture map LOD prior to min-vs-mag determination and mip-level clamping. Assuming mipmapping is enabled, a positive LOD bias will result in a somewhat blurrier image (using less-detailed mip levels) and possibly higher performance, while a negative bias will result in a somewhat crisper image (using more-detailed mip levels) and may lower performance.</p> <div style="text-align: center; background-color: #e6f2ff; padding: 5px; margin-top: 10px;">Programming Notes</div> <p>There is no requirement or need to offset the LOD Bias in order to produce a correct LOD for texture filtering (as was required for correct bilinear and anisotropic filtering in some legacy devices).</p>	Format:	S4.8 2's complement																					
Format:	S4.8 2's complement																									
	0	Anisotropic Algorithm	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>U1 Enumerated Type</td> </tr> </table> <p>Controls which algorithm is used for anisotropic filtering. Generally, the EWA approximation algorithm results in higher image quality than the legacy algorithm.</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>LEGACY</td> <td>Use the legacy algorithm for anisotropic filtering</td> </tr> <tr> <td>1h</td> <td>EWA Approximation</td> <td>Use the new EWA approximation algorithm for anisotropic filtering</td> </tr> </tbody> </table>	Format:	U1 Enumerated Type	Value	Name	Description	0h	LEGACY	Use the legacy algorithm for anisotropic filtering	1h	EWA Approximation	Use the new EWA approximation algorithm for anisotropic filtering												
Format:	U1 Enumerated Type																									
Value	Name	Description																								
0h	LEGACY	Use the legacy algorithm for anisotropic filtering																								
1h	EWA Approximation	Use the new EWA approximation algorithm for anisotropic filtering																								

SAMPLER_STATE											
1	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center; vertical-align: top;">31:20</td> <td> <p>Min LOD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U4.8 in LOD units</td> </tr> </table> <p>Range: [0.0, 14.0], where the upper limit is also bounded by the Max LOD.</p> <p>This field specifies the minimum value used to clamp the computed LOD after LOD bias is applied. Note that the minification-vs.-magnification status is determined after LOD bias and before this maximum (resolution) mip clamping is applied. The integer bits of this field are used to control the "maximum" (highest resolution) mipmap level that may be accessed (where LOD 0 is the highest resolution map). The fractional bits of this value effectively clamp the inter-level trilinear blend factor when trilinear filtering is in use.</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">If Min LOD is greater than Max LOD, Min LOD takes precedence, i.e. the resulting LOD will always be Min LOD.</td> </tr> <tr> <td colspan="2">This field must be zero if the Min or Mag Mode Filter is set to MAPFILTER_MONO</td> </tr> </table> </td> </tr> </table>	31:20	<p>Min LOD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U4.8 in LOD units</td> </tr> </table> <p>Range: [0.0, 14.0], where the upper limit is also bounded by the Max LOD.</p> <p>This field specifies the minimum value used to clamp the computed LOD after LOD bias is applied. Note that the minification-vs.-magnification status is determined after LOD bias and before this maximum (resolution) mip clamping is applied. The integer bits of this field are used to control the "maximum" (highest resolution) mipmap level that may be accessed (where LOD 0 is the highest resolution map). The fractional bits of this value effectively clamp the inter-level trilinear blend factor when trilinear filtering is in use.</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">If Min LOD is greater than Max LOD, Min LOD takes precedence, i.e. the resulting LOD will always be Min LOD.</td> </tr> <tr> <td colspan="2">This field must be zero if the Min or Mag Mode Filter is set to MAPFILTER_MONO</td> </tr> </table>	Format:	U4.8 in LOD units	Programming Notes		If Min LOD is greater than Max LOD, Min LOD takes precedence, i.e. the resulting LOD will always be Min LOD.		This field must be zero if the Min or Mag Mode Filter is set to MAPFILTER_MONO	
31:20	<p>Min LOD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U4.8 in LOD units</td> </tr> </table> <p>Range: [0.0, 14.0], where the upper limit is also bounded by the Max LOD.</p> <p>This field specifies the minimum value used to clamp the computed LOD after LOD bias is applied. Note that the minification-vs.-magnification status is determined after LOD bias and before this maximum (resolution) mip clamping is applied. The integer bits of this field are used to control the "maximum" (highest resolution) mipmap level that may be accessed (where LOD 0 is the highest resolution map). The fractional bits of this value effectively clamp the inter-level trilinear blend factor when trilinear filtering is in use.</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">If Min LOD is greater than Max LOD, Min LOD takes precedence, i.e. the resulting LOD will always be Min LOD.</td> </tr> <tr> <td colspan="2">This field must be zero if the Min or Mag Mode Filter is set to MAPFILTER_MONO</td> </tr> </table>	Format:	U4.8 in LOD units	Programming Notes		If Min LOD is greater than Max LOD, Min LOD takes precedence, i.e. the resulting LOD will always be Min LOD.		This field must be zero if the Min or Mag Mode Filter is set to MAPFILTER_MONO			
Format:	U4.8 in LOD units										
Programming Notes											
If Min LOD is greater than Max LOD, Min LOD takes precedence, i.e. the resulting LOD will always be Min LOD.											
This field must be zero if the Min or Mag Mode Filter is set to MAPFILTER_MONO											
19:8	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center; vertical-align: top;">Max LOD</td> <td> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U4.8 in LOD units</td> </tr> </table> <p>Range: [0.0, 14.0]</p> <p>This field specifies the maximum value used to clamp the computed LOD after LOD bias is applied. Note that the minification-vs.-magnification status is determined after LOD bias and before this minimum (resolution) mip clamping is applied. The integer bits of this field are used to control the "minimum" (lowest resolution) mipmap level that may be accessed. The fractional bits of this value effectively clamp the inter-level trilinear blend factor when trilinear filtering is in use. Force the mip map access to be between the mipmap specified by the integer bits of the Min LOD and the ceiling of the value specified here.</p> </td> </tr> </table>	Max LOD	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U4.8 in LOD units</td> </tr> </table> <p>Range: [0.0, 14.0]</p> <p>This field specifies the maximum value used to clamp the computed LOD after LOD bias is applied. Note that the minification-vs.-magnification status is determined after LOD bias and before this minimum (resolution) mip clamping is applied. The integer bits of this field are used to control the "minimum" (lowest resolution) mipmap level that may be accessed. The fractional bits of this value effectively clamp the inter-level trilinear blend factor when trilinear filtering is in use. Force the mip map access to be between the mipmap specified by the integer bits of the Min LOD and the ceiling of the value specified here.</p>	Format:	U4.8 in LOD units						
Max LOD	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U4.8 in LOD units</td> </tr> </table> <p>Range: [0.0, 14.0]</p> <p>This field specifies the maximum value used to clamp the computed LOD after LOD bias is applied. Note that the minification-vs.-magnification status is determined after LOD bias and before this minimum (resolution) mip clamping is applied. The integer bits of this field are used to control the "minimum" (lowest resolution) mipmap level that may be accessed. The fractional bits of this value effectively clamp the inter-level trilinear blend factor when trilinear filtering is in use. Force the mip map access to be between the mipmap specified by the integer bits of the Min LOD and the ceiling of the value specified here.</p>	Format:	U4.8 in LOD units								
Format:	U4.8 in LOD units										
7	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center; vertical-align: top;">ChromaKey Enable</td> <td> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Enable This field enables the chroma key function.</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">Supported only on a specific subset of surface formats. See section titled: "Surface Formats" in this volume for supported formats. This field must be disabled if min or mag filter is MAPFILTER_MONO or MAPFILTER_ANISOTROPIC. This field must be disabled if used with a surface of type SURFTYPE_3D.</td> </tr> </table> </td> </tr> </table>	ChromaKey Enable	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Enable This field enables the chroma key function.</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">Supported only on a specific subset of surface formats. See section titled: "Surface Formats" in this volume for supported formats. This field must be disabled if min or mag filter is MAPFILTER_MONO or MAPFILTER_ANISOTROPIC. This field must be disabled if used with a surface of type SURFTYPE_3D.</td> </tr> </table>	Format:	Enable This field enables the chroma key function.	Programming Notes		Supported only on a specific subset of surface formats. See section titled: "Surface Formats" in this volume for supported formats. This field must be disabled if min or mag filter is MAPFILTER_MONO or MAPFILTER_ANISOTROPIC. This field must be disabled if used with a surface of type SURFTYPE_3D.			
ChromaKey Enable	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>Enable This field enables the chroma key function.</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">Supported only on a specific subset of surface formats. See section titled: "Surface Formats" in this volume for supported formats. This field must be disabled if min or mag filter is MAPFILTER_MONO or MAPFILTER_ANISOTROPIC. This field must be disabled if used with a surface of type SURFTYPE_3D.</td> </tr> </table>	Format:	Enable This field enables the chroma key function.	Programming Notes		Supported only on a specific subset of surface formats. See section titled: "Surface Formats" in this volume for supported formats. This field must be disabled if min or mag filter is MAPFILTER_MONO or MAPFILTER_ANISOTROPIC. This field must be disabled if used with a surface of type SURFTYPE_3D.					
Format:	Enable This field enables the chroma key function.										
Programming Notes											
Supported only on a specific subset of surface formats. See section titled: "Surface Formats" in this volume for supported formats. This field must be disabled if min or mag filter is MAPFILTER_MONO or MAPFILTER_ANISOTROPIC. This field must be disabled if used with a surface of type SURFTYPE_3D.											
6:5	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center; vertical-align: top;">ChromaKey Index</td> <td> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2</td> </tr> </table> <p>Range: [0, 3]</p> <p>This field specifies the index of the ChromaKey Table entry associated with this Sampler. This field is a "don't care" unless ChromaKey Enable is ENABLED.</p> </td> </tr> </table>	ChromaKey Index	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2</td> </tr> </table> <p>Range: [0, 3]</p> <p>This field specifies the index of the ChromaKey Table entry associated with this Sampler. This field is a "don't care" unless ChromaKey Enable is ENABLED.</p>	Format:	U2						
ChromaKey Index	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2</td> </tr> </table> <p>Range: [0, 3]</p> <p>This field specifies the index of the ChromaKey Table entry associated with this Sampler. This field is a "don't care" unless ChromaKey Enable is ENABLED.</p>	Format:	U2								
Format:	U2										

SAMPLER_STATE

4	ChromaKey Mode	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>U1 Enumerated Type</td> </tr> </table> <p>This field specifies the behavior of the device in the event of a ChromaKey match. This field is ignored if ChromaKey is disabled.</p> <p>KEYFILTER_REPLACE_BLACK: In this mode, each texel that matches the chroma key is replaced with (0,0,0,0) (black with alpha=0) prior to filtering. For YCrCb surface formats, the black value is A=0, R(Cr)=0x80, G(Y)=0x10, B(Cb)=0x80. This will tend to darken/fade edges of keyed regions. Note that the pixel pipeline must be programmed to use the resulting filtered texel value to gain the intended effect, e.g., handle the case of a totally keyed-out region (filtered texel alpha=0) through use of alpha test, etc.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr style="background-color: #e1eef6;"> <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>KEYFILTER_KILL_ON_ANY_MATCH</td> <td>In this mode, if any contributing texel matches the chroma key, the corresponding pixel mask bit for that pixel is cleared. The result of this operation is observable only if the Killed Pixel Mask Return flag is set on the input message.</td> </tr> <tr> <td>1h</td> <td>KEYFILTER_REPLACE_BLACK</td> <td>In this mode, each texel that matches the chroma key is replaced with (0,0,0,0) (black with alpha=0) prior to filtering. For YCrCb surface formats, the black value is A=0, R(Cr)=0x80, G(Y)=0x10, B(Cb)=0x80. This will tend to darken/fade edges of keyed regions. Note that the pixel pipeline must be programmed to use the resulting filtered texel value to gain the intended effect, e.g., handle the case of a totally keyed-out region (filtered texel alpha=0) through use of alpha test, etc.</td> </tr> </tbody> </table>	Format:	U1 Enumerated Type	Value	Name	Description	0h	KEYFILTER_KILL_ON_ANY_MATCH	In this mode, if any contributing texel matches the chroma key, the corresponding pixel mask bit for that pixel is cleared. The result of this operation is observable only if the Killed Pixel Mask Return flag is set on the input message.	1h	KEYFILTER_REPLACE_BLACK	In this mode, each texel that matches the chroma key is replaced with (0,0,0,0) (black with alpha=0) prior to filtering. For YCrCb surface formats, the black value is A=0, R(Cr)=0x80, G(Y)=0x10, B(Cb)=0x80. This will tend to darken/fade edges of keyed regions. Note that the pixel pipeline must be programmed to use the resulting filtered texel value to gain the intended effect, e.g., handle the case of a totally keyed-out region (filtered texel alpha=0) through use of alpha test, etc.							
Format:	U1 Enumerated Type																			
Value	Name	Description																		
0h	KEYFILTER_KILL_ON_ANY_MATCH	In this mode, if any contributing texel matches the chroma key, the corresponding pixel mask bit for that pixel is cleared. The result of this operation is observable only if the Killed Pixel Mask Return flag is set on the input message.																		
1h	KEYFILTER_REPLACE_BLACK	In this mode, each texel that matches the chroma key is replaced with (0,0,0,0) (black with alpha=0) prior to filtering. For YCrCb surface formats, the black value is A=0, R(Cr)=0x80, G(Y)=0x10, B(Cb)=0x80. This will tend to darken/fade edges of keyed regions. Note that the pixel pipeline must be programmed to use the resulting filtered texel value to gain the intended effect, e.g., handle the case of a totally keyed-out region (filtered texel alpha=0) through use of alpha test, etc.																		
3:1	Shadow Function	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>U3 Enumerated Type</td> </tr> </table> <p>This field is used for shadow mapping support via the sample_c message type, and specifies the specific comparison operation to be used. The comparison is between the texture sample red channel (except for alpha-only formats which use the alpha channel), and the "ref" value provided in the input message.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>PREFILTEROP ALWAYS</td> </tr> <tr> <td>1h</td> <td>PREFILTEROP NEVER</td> </tr> <tr> <td>2h</td> <td>PREFILTEROP LESS</td> </tr> <tr> <td>3h</td> <td>PREFILTEROP EQUAL</td> </tr> <tr> <td>4h</td> <td>PREFILTEROP LEQUAL</td> </tr> <tr> <td>5h</td> <td>PREFILTEROP GREATER</td> </tr> <tr> <td>6h</td> <td>PREFILTEROP NOTEQUAL</td> </tr> </tbody> </table>	Format:	U3 Enumerated Type	Value	Name	0h	PREFILTEROP ALWAYS	1h	PREFILTEROP NEVER	2h	PREFILTEROP LESS	3h	PREFILTEROP EQUAL	4h	PREFILTEROP LEQUAL	5h	PREFILTEROP GREATER	6h	PREFILTEROP NOTEQUAL
Format:	U3 Enumerated Type																			
Value	Name																			
0h	PREFILTEROP ALWAYS																			
1h	PREFILTEROP NEVER																			
2h	PREFILTEROP LESS																			
3h	PREFILTEROP EQUAL																			
4h	PREFILTEROP LEQUAL																			
5h	PREFILTEROP GREATER																			
6h	PREFILTEROP NOTEQUAL																			

SAMPLER_STATE		
	7h	PREFILTEROP GEQUAL
0	Cube Surface Control Mode	
	Format:	U1 Enumerated Type
	When sampling from a SURFTYPE_CUBE surface, this field controls whether the TC* Address Control Mode fields are interpreted as programmed or overridden to TEXCOORDMODE_CUBE.	
	Value	Name
	0h	PROGRAMMED
	1h	OVERRIDE
Programming Notes		
This field must be set to CUBECTRLMODE_PROGRAMMED		
2	31:30	Reserved
	29:28	Reserved
	27:26	Reserved
	31:24	Reserved
	25:24	Reserved
	23:6	Indirect State Pointer
	Description	
	This pointer is relative to the Dynamic State Base Address.	
	5	Reserved
	Format:	MBZ
	4	Reserved
	3	Reserved
	2	Reserved
	1	Reserved
0	LOD Clamp Magnification Mode	
	Format:	U1 Enumerated Type
	This field allows the flexibility to control how LOD clamping is handled when in magnification mode.	
	Value	Name
	Description	
0h	MIPNONE	When in magnification mode, Sampler will clamp LOD as if the Mip Mode Filter is MIPFILTER_NONE. This is how OpenGL defines magnification, and therefore it is expected that those drivers would not set this bit.
1h	MIPFILTER	When in magnification mode, Sampler will clamp LOD based on the value of Mip Mode Filter .
3	31:24	Reserved
	23:22	Reduction Type
	Format:	U2 Enumerated Type

SAMPLER_STATE

This field defines the type of reduction that will be performed on the texels in the footprint defined by the **Min/Mag/Mip Filter Mode** fields. This field is ignored if **Reduction Type Enable** is disabled.

Value	Name	Description
0h	STD_FILTER	standard filter
1h	COMPARISON	comparison followed by standard filter
2h	MINIMUM	minimum of footprint
3h	MAXIMUM	maximum of footprint

Programming Notes

The following message types ignore this field: *sample_min*, *sample_max*, *sample_unorm**, *resinfo*, *sampleinfo*, *LOD*, *ld**, *sample_8x8*.

If the current min/mag filter mode is MAPFILTER_MONO, this field is ignored.

The *sample_c*, *sample_l_c*, *sample_d_c*, *sample_b_c*, *gather4_c*, and *gather4_po_c* message types, when used with STD_FILTER, MINIMUM, or MAXIMUM settings of this field, perform the operation of the message of the same name without the "_c". The ref parameter is ignored by hardware.

For message types not listed above, when used with COMPARISON setting of this field, perform the operation of the message of the same name with "_c" included. The ref parameter used by the operation (since it is not delivered in the message) is set to zero.

Restrictions applying to the message whose behavior is being performed must be followed. For example, a sample message used with COMPARISON reduction filter must follow all of the restrictions of *sample_c*. An exception to this is the MINIMUM and MAXIMUM reduction types allow SURFTYPE_1D, 2D, 3D, and CUBE, including with **Surface Array** enabled, even though the *sample_min/max* messages only allow 2D.

Restrictions applying to the message delivered need not be followed. For example, a *sample_c* message used with STD_FILTER reduction filter needs to follow only the restrictions of *sample*, not the restrictions of *sample_c*.

21:19 Maximum Anisotropy

Format: U3 Enumerated Type

This field clamps the maximum value of the anisotropy ratio used by the MAPFILTER_ANISOTROPIC filter (Min or Mag Mode Filter).

Value	Name	Description
0h	RATIO 2:1	At most a 2:1 aspect ratio filter is used
1h	RATIO 4:1	At most a 4:1 aspect ratio filter is used
2h	RATIO 6:1	At most a 6:1 aspect ratio filter is used
3h	RATIO 8:1	At most a 8:1 aspect ratio filter is used
4h	RATIO 10:1	At most a 10:1 aspect ratio filter is used
5h	RATIO 12:1	At most a 12:1 aspect ratio filter is used
6h	RATIO 14:1	At most a 14:1 aspect ratio filter is used

SAMPLER_STATE		
	7h	RATIO 16:1 At most a 16:1 aspect ratio filter is used
18	U Address Mag Filter Rounding Enable	
	Format:	Enable
	Controls whether the texture address is rounded or truncated before being used to select texels to sample. Provides independent control of rounding on one texture address dimension (U/V/R) in either mag or min filter mode.	
	Programming Notes	
	Hardware will not force rounding enable.	
17	U Address Min Filter Rounding Enable	
	Format:	Enable
	Controls whether the texture address is rounded or truncated before being used to select texels to sample. Provides independent control of rounding on one texture address dimension (U/V/R) in either mag or min filter mode.	
	Programming Notes	
	Hardware will not force rounding enable.	
16	V Address Mag Filter Rounding Enable	
	Format:	Enable
	Controls whether the texture address is rounded or truncated before being used to select texels to sample. Provides independent control of rounding on one texture address dimension (U/V/R) in either mag or min filter mode.	
	Programming Notes	
	Hardware will not force rounding enable.	
15	V Address Min Filter Rounding Enable	
	Format:	Enable
	Controls whether the texture address is rounded or truncated before being used to select texels to sample. Provides independent control of rounding on one texture address dimension (U/V/R) in either mag or min filter mode.	
	Programming Notes	
	Hardware will not force rounding enable.	
14	R Address Mag Filter Rounding Enable	
	Format:	Enable
	Controls whether the texture address is rounded or truncated before being used to select texels to sample. Provides independent control of rounding on one texture address dimension (U/V/R) in either mag or min filter mode.	
	Programming Notes	
	Hardware will not force rounding enable.	
13	R Address Min Filter Rounding Enable	
	Format:	Enable
Controls whether the texture address is rounded or truncated before being used to select texels		

SAMPLER_STATE																		
	<p>to sample. Provides independent control of rounding on one texture address dimension (U/V/R) in either mag or min filter mode.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Hardware will not force rounding enable.</td> </tr> </table>	Programming Notes		Hardware will not force rounding enable.														
Programming Notes																		
Hardware will not force rounding enable.																		
12:11	<p>Trilinear Filter Quality</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>U2 Enumerated Type</td> </tr> </table> <p>Selects the quality level for the trilinear filter.</p> <table border="1" style="width: 100%;"> <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>0</td> <td>FULL</td> <td>Full Quality. Both mip maps are sampled under all circumstances.</td> </tr> <tr> <td>1</td> <td>HIGH</td> <td>High Quality. Same as full quality.</td> </tr> <tr> <td>2</td> <td>MED</td> <td>Medium Quality. If the contribution of one mip map is less than 25%, only the other mip map contributes.</td> </tr> <tr> <td>3</td> <td>LOW</td> <td>Low Quality. If the contribution of one mip map is less than 37.5%, only the other mip map contributes.</td> </tr> </tbody> </table>	Format:	U2 Enumerated Type	Value	Name	Description	0	FULL	Full Quality. Both mip maps are sampled under all circumstances.	1	HIGH	High Quality. Same as full quality.	2	MED	Medium Quality. If the contribution of one mip map is less than 25%, only the other mip map contributes.	3	LOW	Low Quality. If the contribution of one mip map is less than 37.5%, only the other mip map contributes.
Format:	U2 Enumerated Type																	
Value	Name	Description																
0	FULL	Full Quality. Both mip maps are sampled under all circumstances.																
1	HIGH	High Quality. Same as full quality.																
2	MED	Medium Quality. If the contribution of one mip map is less than 25%, only the other mip map contributes.																
3	LOW	Low Quality. If the contribution of one mip map is less than 37.5%, only the other mip map contributes.																
10	<p>Non-normalized Coordinate Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field, if enabled, specifies that the input coordinates (U/V/R) are in non-normalized space, where each integer increment is one texel on LOD 0. If disabled, coordinates are normalized, where the range 0 to 1 spans the entire surface.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2"> The following state must be set as indicated if this field is <i>enabled</i>: <ul style="list-style-type: none"> • TCX/Y/Z Address Control Mode must be TEXCOORDMODE_CLAMP, TEXCOORDMODE_HALF_BORDER, or TEXCOORDMODE_CLAMP_BORDER. • Surface Type must be SURFTYPE_2D or SURFTYPE_3D. • Mag Mode Filter must be MAPFILTER_NEAREST or MAPFILTER_LINEAR. • Min Mode Filter must be MAPFILTER_NEAREST or MAPFILTER_LINEAR. • Mip Mode Filter must be MIPFILTER_NONE. • Min LOD must be 0. • Max LOD must be 0. • MIP Count must be 0. • Surface Min LOD must be 0. • Texture LOD Bias must be 0. </td> </tr> </table>	Format:	Enable	Programming Notes		The following state must be set as indicated if this field is <i>enabled</i> : <ul style="list-style-type: none"> • TCX/Y/Z Address Control Mode must be TEXCOORDMODE_CLAMP, TEXCOORDMODE_HALF_BORDER, or TEXCOORDMODE_CLAMP_BORDER. • Surface Type must be SURFTYPE_2D or SURFTYPE_3D. • Mag Mode Filter must be MAPFILTER_NEAREST or MAPFILTER_LINEAR. • Min Mode Filter must be MAPFILTER_NEAREST or MAPFILTER_LINEAR. • Mip Mode Filter must be MIPFILTER_NONE. • Min LOD must be 0. • Max LOD must be 0. • MIP Count must be 0. • Surface Min LOD must be 0. • Texture LOD Bias must be 0. 												
Format:	Enable																	
Programming Notes																		
The following state must be set as indicated if this field is <i>enabled</i> : <ul style="list-style-type: none"> • TCX/Y/Z Address Control Mode must be TEXCOORDMODE_CLAMP, TEXCOORDMODE_HALF_BORDER, or TEXCOORDMODE_CLAMP_BORDER. • Surface Type must be SURFTYPE_2D or SURFTYPE_3D. • Mag Mode Filter must be MAPFILTER_NEAREST or MAPFILTER_LINEAR. • Min Mode Filter must be MAPFILTER_NEAREST or MAPFILTER_LINEAR. • Mip Mode Filter must be MIPFILTER_NONE. • Min LOD must be 0. • Max LOD must be 0. • MIP Count must be 0. • Surface Min LOD must be 0. • Texture LOD Bias must be 0. 																		
9	<p>Reduction Type Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables the Reduction Type field to modify the behavior of messages based on its setting. If this field is disabled, all messages behave as defined and the Reduction Type field is ignored.</p>	Format:	Enable															
Format:	Enable																	

SAMPLER_STATE				
8:6	<p>TCX Address Control Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>Texture Coordinate Mode Enumerated Type</td> </tr> </table> <p>Controls how the 1st (TCX, aka U) component of input texture coordinates are mapped to texture map addresses - specifically, how coordinates "outside" the texture are handled (wrap/clamp/mirror). The setting of this field is subject to being overridden by the Cube Surface Control Mode field when sampling from a SURFTYPE_CUBE surface.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>When using cube map texture coordinates, each TC component must have the same Address Control Mode.</p> <p>When TEXCOORDMODE_CUBE is not used accessing a cube map, the map's Cube Face Enable field must be programmed to 111111b (all faces enabled).</p> <p>MAPFILTER_MONO: Texture addressing modes must all be set to TEXCOORDMODE_CLAMP_BORDER. The Border Color is ignored in this mode, a constant value of 0 is used for border color. Software must pad the border texels within the map itself with 0.</p> <p>If Surface Format is PLANAR*, this field must be set to TEXCOORDMODE_CLAMP.</p>	Format:	Texture Coordinate Mode Enumerated Type	Programming Notes
Format:	Texture Coordinate Mode Enumerated Type			
Programming Notes				
5:3	<p>TCY Address Control Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>Texture Coordinate Mode Enumerated Type</td> </tr> </table> <p>Controls how the 2nd (TCY, aka V) component of input texture coordinates are mapped to texture map addresses - specifically, how coordinates "outside" the texture are handled (wrap/clamp/mirror). See Address TCX Control Mode above for details</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>If this field is set to TEXCOORDMODE_CLAMP_BORDER or TEXCOORDMODE_HALF_BORDER and a 1D surface is sampled, incorrect blending with the border color in the vertical direction may occur.</p>	Format:	Texture Coordinate Mode Enumerated Type	Programming Notes
Format:	Texture Coordinate Mode Enumerated Type			
Programming Notes				
2:0	<p>TCZ Address Control Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>Texture Coordinate Mode Enumerated Type</td> </tr> </table> <p>Controls how the 3rd (TCZ) component of input texture coordinates are mapped to texture map addresses - specifically, how coordinates "outside" the texture are handled (wrap/clamp/mirror). See Address TCX Control Mode above for details</p>	Format:	Texture Coordinate Mode Enumerated Type	
Format:	Texture Coordinate Mode Enumerated Type			

SCISSOR_RECT

SCISSOR_RECT								
Source:	RenderCS							
Size (in bits):	64							
Default Value:	0x00000000, 0x00000000							
<p>The viewport-specific state used by the SF unit (SCISSOR_RECT) is stored as an array of up to 16 elements, each of which contains the DWords described below. The start of each element is spaced 2 DWords apart. The location of first element of the array, as specified by Pointer to SCISSOR_RECT, is aligned to a 32-byte boundary.</p>								
DWord	Bit	Description						
0	31:16	<p>Scissor Rectangle Y Min</p> <table border="1"> <tr> <td>Format:</td> <td>U16 Pixels from Drawing Rectangle origin (upper left corner)</td> </tr> </table> <p>Specifies Y Min coordinate of (inclusive) Scissor Rectangle used for scissor test. Pixels with (Draw Rectangle-relative) Y coordinates less than Y Min will be clipped out if Scissor Rectangle is enabled. NOTE: If Y Min is set to a value greater than Y Max, all primitives will be discarded for this viewport.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Format:	U16 Pixels from Drawing Rectangle origin (upper left corner)	Value	Name	[0,16383]	
		Format:	U16 Pixels from Drawing Rectangle origin (upper left corner)					
Value	Name							
[0,16383]								
15:0	15:0	<p>Scissor Rectangle X Min</p> <table border="1"> <tr> <td>Format:</td> <td>U16 Pixels from Drawing Rectangle origin (upper left corner)</td> </tr> </table> <p>Specifies X Min coordinate of (inclusive) Scissor Rectangle used for scissor test. Pixels with (Draw Rectangle-relative) X coordinates less than X Min will be clipped out if Scissor Rectangle is enabled. NOTE: If X Min is set to a value greater than X Max, all primitives will be discarded for this viewport.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Format:	U16 Pixels from Drawing Rectangle origin (upper left corner)	Value	Name	[0,16383]	
		Format:	U16 Pixels from Drawing Rectangle origin (upper left corner)					
Value	Name							
[0,16383]								
1	31:16	<p>Scissor Rectangle Y Max</p> <table border="1"> <tr> <td>Format:</td> <td>U16 Pixels from Drawing Rectangle origin (upper left corner)</td> </tr> </table> <p>Specifies Y Max coordinate of (inclusive) Scissor Rectangle used for scissor test. Pixels with (Draw Rectangle-relative) Y coordinates greater than Y Max will be clipped out if Scissor Rectangle is enabled.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Format:	U16 Pixels from Drawing Rectangle origin (upper left corner)	Value	Name	[0,16383]	
		Format:	U16 Pixels from Drawing Rectangle origin (upper left corner)					
	Value	Name						
	[0,16383]							
15:0	15:0	<p>Scissor Rectangle X Max</p> <table border="1"> <tr> <td>Format:</td> <td>U16 Pixels from Drawing Rectangle origin (upper left corner)</td> </tr> </table> <p>Specifies X Max coordinate of (inclusive) Scissor Rectangle used for scissor test. Pixels with (Draw Rectangle-relative) Y coordinates greater than X Max will be clipped out if Scissor Rectangle is enabled.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0-16383</td> <td></td> </tr> </tbody> </table>	Format:	U16 Pixels from Drawing Rectangle origin (upper left corner)	Value	Name	0-16383	
		Format:	U16 Pixels from Drawing Rectangle origin (upper left corner)					
Value	Name							
0-16383								

Scratch Hword Block Message Header

MH_A32_HWB - Scratch Hword Block Message Header				
Source:	DataPort 0			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0-2	95:0	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td>Ignore</td> </tr> </table> Ignored	Format:	Ignore
Format:	Ignore			
3	31:0	Per Thread Scratch Space <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="color: red;">MHC_PTSS</td> </tr> </table> Specifies amount of scratch space used by this thread, for Stateless bounds checking.	Format:	MHC_PTSS
Format:	MHC_PTSS			
4	31:0	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td>Ignore</td> </tr> </table> Ignored	Format:	Ignore
Format:	Ignore			
5	31:0	Buffer Base Address <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="color: red;">MHC_A32_BBA</td> </tr> </table> Specifies the surface address offset page [31:10] for A32 stateless messages.	Format:	MHC_A32_BBA
Format:	MHC_A32_BBA			
6-7	63:0	Reserved <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td>Ignore</td> </tr> </table> Ignored	Format:	Ignore
Format:	Ignore			

SF_CLIP_VIEWPORT

SF_CLIP_VIEWPORT				
Source:	RenderCS			
Size (in bits):	512			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	Viewport Matrix Element m00 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Format:	IEEE_Float
Format:	IEEE_Float			
1	31:0	Viewport Matrix Element m11 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Format:	IEEE_Float
Format:	IEEE_Float			
2	31:0	Viewport Matrix Element m22 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Format:	IEEE_Float
Format:	IEEE_Float			
3	31:0	Viewport Matrix Element m30 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Format:	IEEE_Float
Format:	IEEE_Float			
4	31:0	Viewport Matrix Element m31 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Format:	IEEE_Float
Format:	IEEE_Float			
5	31:0	Viewport Matrix Element m32 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table>	Format:	IEEE_Float
Format:	IEEE_Float			
6	31:0	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
7	31:0	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
8	31:0	X Min Clip Guardband <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table> <p>. This 32-bit float represents the XMin guardband boundary (normalized to Viewport.XMin == -1.0f). This corresponds to the left boundary of the NDC guardband.</p>	Format:	IEEE_Float
Format:	IEEE_Float			
9	31:0	X Max Clip Guardband <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table> <p>This 32-bit float represents the XMax guardband boundary (normalized to Viewport.XMax == 1.0f). This corresponds to the right boundary of the NDC guardband.</p>	Format:	IEEE_Float
Format:	IEEE_Float			
10	31:0	Y Min Clip Guardband <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>IEEE_Float</td> </tr> </table> <p>This 32-bit float represents the YMin guardband boundary (normalized to Viewport.YMin == -1.0f). This corresponds to the bottom boundary of the NDC guardband.</p>	Format:	IEEE_Float
Format:	IEEE_Float			

SF_CLIP_VIEWPORT		
11	31:0	Y Max Clip Guardband
		Format: IEEE_Float
		This 32-bit float represents the YMax guardband boundary (normalized to Viewport.YMax == 1.0f). This corresponds to the top boundary of the NDC guardband.
12	31:0	X Min ViewPort
		Format: IEEE_Float
		This 32-bit float represents the Viewport.XMin. This is the X min of the viewport extents as programmed by API, and this value should be programmed in Screen Space coordinate and not as normalized coordinate.
13	31:0	X Max ViewPort
		Format: IEEE_Float
		This 32-bit float represents the Viewport.XMax. This is the X max of the viewport extents as programmed by API, and this value should be programmed in Screen Space coordinate and not as normalized coordinate.
14	31:0	Y Min ViewPort
		Format: IEEE_Float
		This 32-bit float represents the Viewport.YMin. This is the Y min of the viewport extents as programmed by API, and this value should be programmed in Screen Space coordinate and not as normalized coordinate.
15	31:0	Y Max ViewPort
		Format: IEEE_Float
		This 32-bit float represents the Viewport.Ymax. This is the Y max of the viewport extents as programmed by API, and this value should be programmed in Screen Space coordinate and not as normalized coordinate.

SF_OUTPUT_ATTRIBUTE_DETAIL

SF_OUTPUT_ATTRIBUTE_DETAIL				
Source:	RenderCS			
Size (in bits):	16			
Default Value:	0x00000000			
DWord	Bit	Description		
0	15	<p>Component Override W</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If set, the W component of this output Attribute is overridden by the W component of the constant vector specified by ConstantSource.</p>	Format:	Enable
	Format:	Enable		
	14	<p>Component Override Z</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If set, the Z component of this output Attribute is overridden by the Z component of the constant vector specified by ConstantSource.</p>	Format:	Enable
	Format:	Enable		
	13	<p>Component Override Y</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If set, the Y component of output Attribute is overridden by the Y component of the constant vector specified by ConstantSource.</p>	Format:	Enable
Format:	Enable			
12	<p>Component Override X</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If set, the X component of output Attribute is overridden by the X component of the constant vector specified by ConstantSource.</p>	Format:	Enable	
Format:	Enable			
11	<p>Swizzle Control Mode</p> <table border="1"> <tr> <td>Format:</td> <td>U1 Enumerated Type</td> </tr> </table> <p>When Attribute Swizzle Enable is ENABLED, this bit controls whether attributes 0-15 or 16-31 are subject to the following swizzle controls:</p> <ul style="list-style-type: none"> • Component Override X/Y/Z/W • Constant Source • Swizzle Select • Source Attribute • WrapShortest Enables <p>Note that the Number of SF Output Attributes field specifies how many attributes are output. Note: This field does not impact any functions which provide separate states for all 32 attributes (e.g., Point sprite, Constant interpolation). Note: This field is only valid for the first indexed attribute (Attribute[0]). For all other indices, it is Reserved and MBZ.</p>	Format:	U1 Enumerated Type	
Format:	U1 Enumerated Type			

SF_OUTPUT_ATTRIBUTE_DETAIL			
10:9	Constant Source		
	Format:	U2 enumerated type	
	This state selects a constant vector which can be used to override individual components of this Attribute		
	Value	Name	Description
	0h	CONST_0000	Constant.xyzw = 0.0,0.0,0.0,0.0
	1h	CONST_0001_FLOAT	Constant.xyzw = 0.0,0.0,0.0,1.0
	2h	CONST_1111_FLOAT	Constant.xyzw = 1.0,1.0,1.0,1.0
	3h	PRIM_ID	Constant.xyzw = PrimID (replicated)
	8	Reserved	
		Format:	MBZ
7:6	Swizzle Select		
	Format:	U2 enumerated type	
	This state, along with Source Attribute, specifies the source for this output Attribute.		
	Value	Name	Description
	0h	INPUTATTR	This attribute is sourced from AttrInputReg[SourceAttribute]
	1h	INPUTATTR_FACING	If the object is front-facing, this attribute is sourced from AttrInputReg[SourceAttribute]. If the object is back-facing, this attribute is sourced from AttrInputReg[SourceAttribute+1].
5	Reserved		
	Format:	MBZ	
	4:0	Source Attribute	
		Format:	U5
This field selects the source attribute for this Attribute. Source attribute 0 corresponds to the first 128 bits of data indicated by Vertex URB Entry Read Offset			

SFC_8x8_AVS_COEFFICIENTS

SFC_8x8_AVS_COEFFICIENTS			
Source:	BSpec		
Size (in bits):	256		
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000		
Description			
ExistsIf = AVS && (Function_mode = 0)			
DWord	Bit	Description	
0	31:24	ZeroYFilterCoefficient1 Format: <table border="1" style="display: inline-table;"><tr><td>S1.6 2's Complement</td></tr></table> Range: [-2, +2)	S1.6 2's Complement
	S1.6 2's Complement		
	23:16	ZeroXFilterCoefficient1 Format: <table border="1" style="display: inline-table;"><tr><td>S1.6 2's Complement</td></tr></table> Range: [-2, +2)	S1.6 2's Complement
	S1.6 2's Complement		
15:8	ZeroYFilterCoefficient0 Format: <table border="1" style="display: inline-table;"><tr><td>S1.6 2's Complement</td></tr></table> Range: [-2, +2)	S1.6 2's Complement	
S1.6 2's Complement			
7:0	ZeroXFilterCoefficient0 Format: <table border="1" style="display: inline-table;"><tr><td>S1.6 2's Complement</td></tr></table> Range: [-2, +2)	S1.6 2's Complement	
S1.6 2's Complement			
1	31:24	ZeroYFilterCoefficient3 Format: <table border="1" style="display: inline-table;"><tr><td>S1.6 2's Complement</td></tr></table> Range: [-2, +2)	S1.6 2's Complement
	S1.6 2's Complement		
	23:16	ZeroXFilterCoefficient3 Format: <table border="1" style="display: inline-table;"><tr><td>S1.6 2's Complement</td></tr></table> Range: [-2, +2)	S1.6 2's Complement
S1.6 2's Complement			
15:8	ZeroYFilterCoefficient2 Format: <table border="1" style="display: inline-table;"><tr><td>S1.6 2's Complement</td></tr></table> Range: [-2, +2)	S1.6 2's Complement	
S1.6 2's Complement			

SFC_8x8_AVS_COEFFICIENTS				
	7:0	ZeroXFilterCoefficient2 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement
Format:	S1.6 2's Complement			
2	31:24	ZeroYFilterCoefficient5 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement
	Format:	S1.6 2's Complement		
	23:16	ZeroXFilterCoefficient5 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement
	Format:	S1.6 2's Complement		
15:8	ZeroYFilterCoefficient4 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement	
Format:	S1.6 2's Complement			
7:0	ZeroXFilterCoefficient4 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement	
Format:	S1.6 2's Complement			
3	31:24	ZeroYFilterCoefficient7 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement
	Format:	S1.6 2's Complement		
	23:16	ZeroXFilterCoefficient7 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement
	Format:	S1.6 2's Complement		
15:8	ZeroYFilterCoefficient6 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement	
Format:	S1.6 2's Complement			
7:0	ZeroXFilterCoefficient6 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement	
Format:	S1.6 2's Complement			
4	31:24	OneXFilterCoefficient3 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2.0, +2.0)	Format:	S1.6 2's Complement
Format:	S1.6 2's Complement			

SFC_8x8_AVS_COEFFICIENTS		
	23:16	OneXFilterCoefficient2 Format: S1.6 2's Complement Range: [-1.0, +1.0)
	15:0	Reserved Format: MBZ
5	31:16	Reserved Format: MBZ
	15:8	OneXFilterCoefficient5 Format: S1.6 2's Complement Range: [-1.0, +1.0)
	7:0	OneXFilterCoefficient4 Format: S1.6 2's Complement Range: [-2.0, +2.0)
6	31:24	OneYFilterCoefficient3 Format: S1.6 2's Complement Range: [-2.0, +2.0)
	23:16	OneYFilterCoefficient2 Format: S1.6 2's Complement Range: [-1.0, +1.0)
	15:0	Reserved Format: MBZ
7	31:16	Reserved Format: MBZ
	15:8	OneYFilterCoefficient5 Format: S1.6 2's Complement Range: [-1.0, +1.0)
	7:0	OneYFilterCoefficient4 Format: S1.6 2's Complement Range: [-2.0, +2.0)

SIMD4x2 MSAA Typed Surface 32-Bit Address Payload

MAP32B_MSAA_TS_SIMD4X2 - SIMD4x2 MSAA Typed Surface 32-Bit Address Payload				
Source:	BSpec			
Size (in bits):	512			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.0	31:0	<p>Slot0 Sample Number</p> <table border="1"> <tr> <td>Format:</td> <td>MACD_MSAA_SN</td> </tr> </table> <p>Specifies the sample number for slot 0</p>	Format:	MACD_MSAA_SN
Format:	MACD_MSAA_SN			
0.1-0.3	95:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
0.4-0.4	31:0	<p>Slot1 Sample Number</p> <table border="1"> <tr> <td>Format:</td> <td>MACD_MSAA_SN</td> </tr> </table> <p>Specifies the sample number for slot 1</p>	Format:	MACD_MSAA_SN
Format:	MACD_MSAA_SN			
0.5-0.7	95:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>Ignored</p>	Format:	MBZ
Format:	MBZ			
1.0-1.0	31:0	<p>U0</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the U channel address offset for slot 0.</p>	Format:	U32
Format:	U32			
1.1-1.1	31:0	<p>V0</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the V channel address offset for slot 0.</p>	Format:	U32
Format:	U32			
1.2-1.2	31:0	<p>R0</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the R channel address offset for slot 0.</p>	Format:	U32
Format:	U32			

MAP32B_MSAA_TS_SIMD4X2 - SIMD4x2 MSAA Typed Surface 32-Bit Address Payload				
1.3-1.3	31:0	<p>LOD0</p> <table border="1"> <tr> <td>Format:</td> <td>MACD_LOD</td> </tr> </table> <p>Specifies the LOD for slot 0.</p>	Format:	MACD_LOD
Format:	MACD_LOD			
1.4-1.4	31:0	<p>U1</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the U channel address offset for slot 1.</p>	Format:	U32
Format:	U32			
1.5-1.5	31:0	<p>V1</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the V channel address offset for slot 1.</p>	Format:	U32
Format:	U32			
1.6-1.6	31:0	<p>R1</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the R channel address offset for slot 1.</p>	Format:	U32
Format:	U32			
1.7-1.7	31:0	<p>LOD1</p> <table border="1"> <tr> <td>Format:</td> <td>MACD_LOD</td> </tr> </table> <p>Specifies the LOD for slot 1.</p>	Format:	MACD_LOD
Format:	MACD_LOD			

SIMD4x2 Typed Surface 32-Bit Address Payload

MAP32B_TS_SIMD4X2 - SIMD4x2 Typed Surface 32-Bit Address Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	U0
		Format: U32 Specifies the U channel address offset for slot 0.
1	31:0	V0
		Format: U32 Specifies the V channel address offset for slot 0.
2	31:0	R0
		Format: U32 Specifies the R channel address offset for slot 0.
3	31:0	LOD0
		Format: MACD_LOD Specifies the LOD for slot 0.
4	31:0	U1
		Format: U32 Specifies the U channel address offset for slot 1.
5	31:0	V1
		Format: U32 Specifies the V channel address offset for slot 1.
6	31:0	R1
		Format: U32 Specifies the R channel address offset for slot 1.
7	31:0	LOD1
		Format: MACD_LOD Specifies the LOD for slot 1.

SIMD4x2 Untyped BUFFER Surface 32-Bit Address Payload

MAP32B_USU_SIMD4X2 - SIMD4x2 Untyped BUFFER Surface 32-Bit Address Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	<p>U0</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the U channel address offset for slot 0.</p>	Format:	U32
Format:	U32			
1-3	95:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
4	31:0	<p>U1</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the U channel address offset for slot 1.</p>	Format:	U32
Format:	U32			
5-7	95:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

SIMD4x2 Untyped BUFFER Surface 64-Bit Address Payload

MAP64B_USU_SIMD4X2 - SIMD4x2 Untyped BUFFER Surface 64-Bit Address Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0-1	63:0	<p>U0</p> <table border="1"> <tr> <td>Format:</td> <td>U64</td> </tr> </table> <p>Specifies the U channel address offset for slot 0.</p>	Format:	U64
Format:	U64			
2-3	63:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
4-5	63:0	<p>U1</p> <table border="1"> <tr> <td>Format:</td> <td>U64</td> </tr> </table> <p>Specifies the U channel address offset for slot 1.</p>	Format:	U64
Format:	U64			
6-7	63:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

SIMD4x2 Untyped STRBUF Surface 32-Bit Address Payload

MAP32B_USUV_SIMD4X2 - SIMD4x2 Untyped STRBUF Surface 32-Bit Address Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	<p>U0</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the U channel address offset for slot 0.</p>	Format:	U32
Format:	U32			
1	31:0	<p>V0</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the V channel address offset for slot 0.</p>	Format:	U32
Format:	U32			
2-3	63:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
4	31:0	<p>U1</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the U channel address offset for slot 1.</p>	Format:	U32
Format:	U32			
5	31:0	<p>V1</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the V channel address offset for slot 1.</p>	Format:	U32
Format:	U32			
6-7	63:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

SIMD4x2 32-Bit Address Payload

MAP32B_SIMD4X2 - SIMD4x2 32-Bit Address Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	<p>Offset0</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the address offset for slot 0.</p>	Format:	U32
Format:	U32			
1-3	95:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
4	31:0	<p>Offset1</p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the address offset for slot 1.</p>	Format:	U32
Format:	U32			
5-7	95:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

MDP_RTW_8DS - SIMD8 Dual Source Render Target Data Payload		
6.0-6.7	255:0	Src1 Blue
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots[7:0] or [15:8] of Src1 Blue
Format:	MDP_DW_SIMD8	
7.0-7.7	255:0	Src1 Alpha
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots[7:0] or [15:8] of Src1 Alpha
Format:	MDP_DW_SIMD8	

SIMD8 LOD Message Address Payload Control

MACR_LOD_SIMD8 - SIMD8 LOD Message Address Payload Control		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0	31:0	Slot0 LOD Format: MACD_LOD Specifies the LOD for slot 0
0.1	31:0	Slot1 LOD Format: MACD_LOD Specifies the LOD for slot 1
0.2	31:0	Slot2 LOD Format: MACD_LOD Specifies the LOD for slot 2
0.3	31:0	Slot3 LOD Format: MACD_LOD Specifies the LOD for slot 3
0.4	31:0	Slot4 LOD Format: MACD_LOD Specifies the LOD for slot 4
0.5	31:0	Slot5 LOD Format: MACD_LOD Specifies the LOD for slot 5
0.6	31:0	Slot6 LOD Format: MACD_LOD Specifies the LOD for slot 6
0.7	31:0	Slot7 LOD Format: MACD_LOD Specifies the LOD for slot 7

MAP32B_MSAА_TS_SIMD8 - SIMD8 MSAА Typed Surface 32-Bit Address Payload

0.7	31:0	<p>Slot7 Sample Number</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;">MACD_MSAА_SN</td> </tr> </table> <p>Specifies the sample number for slot 7</p>	Format:	MACD_MSAА_SN
Format:	MACD_MSAА_SN			
1.0-1.7	255:0	<p>U</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;">MACR_32b</td> </tr> </table> <p>Specifies the U channel for slots [7:0]</p>	Format:	MACR_32b
Format:	MACR_32b			
2.0-2.7	255:0	<p>V</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;">MACR_32b</td> </tr> </table> <p>Specifies the V channel for slots [7:0]</p>	Format:	MACR_32b
Format:	MACR_32b			
3.0-3.7	255:0	<p>R</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;">MACR_32b</td> </tr> </table> <p>Specifies the R channel for slots [7:0]</p>	Format:	MACR_32b
Format:	MACR_32b			
4.0-4.7	255:0	<p>LOD</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;">MACR_LOD_SIMD8</td> </tr> </table> <p>Specifies the LOD for slots [7:0]</p>	Format:	MACR_LOD_SIMD8
Format:	MACR_LOD_SIMD8			

SIMD8 Render Target Data Payload

MDP_RTW_8 - SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1024	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Red Format: MDP_DW_SIMD8 Slots [7:0] Red
1.0-1.7	255:0	Green Format: MDP_DW_SIMD8 Slots [7:0] Green
2.0-2.7	255:0	Blue Format: MDP_DW_SIMD8 Slots [7:0] Blue
3.0-3.7	255:0	Alpha Format: MDP_DW_SIMD8 Slots [7:0] Alpha

SIMD8 Untyped BUFFER Surface 32-Bit Address Payload

MAP32B_USU_SIMD8 - SIMD8 Untyped BUFFER Surface 32-Bit Address Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	U
		<table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the U channel for slots [7:0]</p>
Format:	MACR_32b	

SIMD8 Untyped BUFFER Surface 64-Bit Address Payload

MAP64B_USU_SIMD8 - SIMD8 Untyped BUFFER Surface 64-Bit Address Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	U3_U0
		<table border="1"> <tr> <td>Format:</td> <td>MACR_64b</td> </tr> </table> <p>Specifies the U channel for slots [3:0]</p>
Format:	MACR_64b	
1.0-1.7	255:0	U7_U4
		<table border="1"> <tr> <td>Format:</td> <td>MACR_64b</td> </tr> </table> <p>Specifies the U channel for slots [7:4]</p>
Format:	MACR_64b	

SIMD8 Untyped STRBUF Surface 32-Bit Address Payload

MAP32B_USUV_SIMD8 - SIMD8 Untyped STRBUF Surface 32-Bit Address Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	U
		<table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the U channel for slots [7:0]</p>
Format:	MACR_32b	
1.0-1.7	255:0	V
		<table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the V channel for slots [7:0]</p>
Format:	MACR_32b	

MDP_RTW_16 - SIMD16 Render Target Data Payload		
6.0-6.7	255:0	Alpha[7:0]
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Alpha
Format:	MDP_DW_SIMD8	
7.0-7.7	255:0	Alpha[15:7]
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:7] Alpha
Format:	MDP_DW_SIMD8	

SIMD16 Untyped BUFFER Surface 32-Bit Address Payload

MAP32B_USU_SIMD16 - SIMD16 Untyped BUFFER Surface 32-Bit Address Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	U[7:0]
		<table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the U channel for slots [7:0]</p>
Format:	MACR_32b	
1.0-1.7	255:0	U[15:8]
		<table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the U channel for slots [15:8]</p>
Format:	MACR_32b	

SIMD16 Untyped STRBUF Surface 32-Bit Address Payload

MAP32B_USUV_SIMD16 - SIMD16 Untyped STRBUF Surface 32-Bit Address Payload				
Source:	BSpec			
Size (in bits):	1024			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<p>U7_U0</p> <table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the U channel for slots [7:0]</p>	Format:	MACR_32b
Format:	MACR_32b			
1.0-1.7	255:0	<p>U15_U8</p> <table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the U channel for slots [15:8]</p>	Format:	MACR_32b
Format:	MACR_32b			
2.0-2.7	255:0	<p>V7_V0</p> <table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the V channel for slots [7:0]</p>	Format:	MACR_32b
Format:	MACR_32b			
3.0-3.7	255:0	<p>V15_V8</p> <table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the V channel for slots [15:8]</p>	Format:	MACR_32b
Format:	MACR_32b			

SIMD 32-Bit Address Payload Control

MACR_32B - SIMD 32-Bit Address Payload Control		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0	31:0	Offset0
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U32</td></tr></table> Specifies the address offset for slot 0 in this payload register.
	U32	
0.1	31:0	Offset1
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U32</td></tr></table> Specifies the address offset for slot 1 in this payload register.
	U32	
0.2	31:0	Offset2
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U32</td></tr></table> Specifies the address offset for slot 2 in this payload register.
	U32	
0.3	31:0	Offset3
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U32</td></tr></table> Specifies the address offset for slot 3 in this payload register.
	U32	
0.4	31:0	Offset4
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U32</td></tr></table> Specifies the address offset for slot 4 in this payload register.
	U32	
0.5	31:0	Offset5
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U32</td></tr></table> Specifies the address offset for slot 5 in this payload register.
	U32	
0.6	31:0	Offset6
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U32</td></tr></table> Specifies the address offset for slot 6 in this payload register.
	U32	
0.7	31:0	Offset7
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U32</td></tr></table> Specifies the address offset for slot 7 in this payload register.
	U32	

SIMD 64-Bit Address Payload Control

MACR_64B - SIMD 64-Bit Address Payload Control		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.1	63:0	Offset0
		Format: U64 Specifies the address offset for slot 0 in this payload register.
0.2-0.3	63:0	Offset1
		Format: U64 Specifies the address offset for slot 1 in this payload register.
0.4-0.5	63:0	Offset2
		Format: U64 Specifies the address offset for slot 2 in this payload register.
0.6-0.7	63:0	Offset3
		Format: U64 Specifies the address offset for slot 3 in this payload register.

SIMD8 32-Bit Address Payload

MAP32B_SIMD8 - SIMD8 32-Bit Address Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<p>Offset[7:0]</p> <table border="1"> <tr> <td>Format:</td> <td>MACR_32b</td> </tr> </table> <p>Specifies the address offset for Slots [7:0].</p>	Format:	MACR_32b
Format:	MACR_32b			

SIMD8 64-Bit Address Payload

MAP64B_SIMD8 - SIMD8 64-Bit Address Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Offset[3:0]
		<table border="1"> <tr> <td>Format:</td> <td>MACR_64b</td> </tr> </table> <p>Specifies the address offset for slots [3:0].</p>
Format:	MACR_64b	
1.0-1.7	255:0	Offset[7:4]
		<table border="1"> <tr> <td>Format:</td> <td>MACR_64b</td> </tr> </table> <p>Specifies the address offset for slots [7:4].</p>
Format:	MACR_64b	

SIMD16 32-Bit Address Payload

MAP32B_SIMD16 - SIMD16 32-Bit Address Payload		
Source:	BSpec	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Offset[7:0]
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MACR_32b</td> </tr> </table> <p>Specifies the address offset for slots [7:0].</p>
Format:	MACR_32b	
1.0-1.7	255:0	Offset[15:8]
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">MACR_32b</td> </tr> </table> <p>Specifies the address offset for slots [15:8].</p>
Format:	MACR_32b	

SIMD Mode 2 Message Descriptor Control Field

MDC_SM2 - SIMD Mode 2 Message Descriptor Control Field																	
Source:	BSpec																
Size (in bits):	1																
Default Value:	0x00000000																
DWord	Bit	Description															
0	0	<p>SIMD Mode</p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">Enumeration</td> </tr> <tr> <td colspan="3">Specifies the SIMD mode of the message (number of slots processed)</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>00h</td> <td>SIMD8</td> <td>SIMD8</td> </tr> <tr> <td>01h</td> <td>SIMD16</td> <td>SIMD16</td> </tr> </table>	Format:	Enumeration		Specifies the SIMD mode of the message (number of slots processed)			Value	Name	Description	00h	SIMD8	SIMD8	01h	SIMD16	SIMD16
Format:	Enumeration																
Specifies the SIMD mode of the message (number of slots processed)																	
Value	Name	Description															
00h	SIMD8	SIMD8															
01h	SIMD16	SIMD16															

SIMD Mode 3 Message Descriptor Control Field

MDC_SM3 - SIMD Mode 3 Message Descriptor Control Field																					
Source:	BSpec																				
Size (in bits):	2																				
Default Value:	0x00000000																				
DWord	Bit	Description																			
0	1:0	<p>SIMD Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> <tr> <td colspan="2">Specifies the SIMD mode of the message (number of slots processed)</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">SIMD4x2</td> <td style="text-align: center;">SIMD4x2</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">SIMD16</td> <td style="text-align: center;">SIMD16</td> </tr> <tr> <td style="text-align: center;">02h</td> <td style="text-align: center;">SIMD8</td> <td style="text-align: center;">SIMD8</td> </tr> <tr> <td style="text-align: center;">03h</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Ignored</td> </tr> </table>	Format:	Enumeration	Specifies the SIMD mode of the message (number of slots processed)		Value	Name	Description	00h	SIMD4x2	SIMD4x2	01h	SIMD16	SIMD16	02h	SIMD8	SIMD8	03h	Reserved	Ignored
Format:	Enumeration																				
Specifies the SIMD mode of the message (number of slots processed)																					
Value	Name	Description																			
00h	SIMD4x2	SIMD4x2																			
01h	SIMD16	SIMD16																			
02h	SIMD8	SIMD8																			
03h	Reserved	Ignored																			

SLM Surface Pixel Mask Message Header

MH1_SLM_PSM - SLM Surface Pixel Mask Message Header				
Source:	DataPort 1			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x0000FFFF			
DWord	Bit	Description		
0-6	223:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
7	31:0	<p>Pixel Sample Mask</p> <table border="1"> <tr> <td>Format:</td> <td>MHC_PSM</td> </tr> </table> <p>Specifies the 16-bit Pixel/Sample Mask used with SIMD16 and SIMD8 surfaces.</p>	Format:	MHC_PSM
Format:	MHC_PSM			

Slot Group 2 Message Descriptor Control Field

MDC_SG2 - Slot Group 2 Message Descriptor Control Field															
Source:	BSpec														
Size (in bits):	1														
Default Value:	0x00000000														
DWord	Bit	Description													
0	0	<p>SIMD Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> <tr> <td colspan="2">Controls which 8 bits of Pixel/Sample Mask in the message header are ANDed with the execution mask to determine which slots are accessed. This field is ignored if the header is not present.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">SG8L</td> <td style="text-align: center;">Use low 8 slots</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">SG8U</td> <td style="text-align: center;">Use high 8 slots</td> </tr> </table>	Format:	Enumeration	Controls which 8 bits of Pixel/Sample Mask in the message header are ANDed with the execution mask to determine which slots are accessed. This field is ignored if the header is not present.		Value	Name	Description	00h	SG8L	Use low 8 slots	01h	SG8U	Use high 8 slots
Format:	Enumeration														
Controls which 8 bits of Pixel/Sample Mask in the message header are ANDed with the execution mask to determine which slots are accessed. This field is ignored if the header is not present.															
Value	Name	Description													
00h	SG8L	Use low 8 slots													
01h	SG8U	Use high 8 slots													

Slot Group 3 Message Descriptor Control Field

MDC_SG3 - Slot Group 3 Message Descriptor Control Field																					
Source:	BSpec																				
Size (in bits):	2																				
Default Value:	0x00000000																				
DWord	Bit	Description																			
0	1:0	<p>SIMD Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> <tr> <td colspan="2">Controls which 8 bits of Pixel/Sample Mask in the message header are ANDed with the execution mask to determine which slots are accessed. This field is ignored if the header is not present.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">SG4x2</td> <td style="text-align: center;">SIMD4x2</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">SG8L</td> <td style="text-align: center;">Use low 8 slots</td> </tr> <tr> <td style="text-align: center;">02h</td> <td style="text-align: center;">SG8U</td> <td style="text-align: center;">Use high 8 slots</td> </tr> <tr> <td style="text-align: center;">03h</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Ignored</td> </tr> </table>	Format:	Enumeration	Controls which 8 bits of Pixel/Sample Mask in the message header are ANDed with the execution mask to determine which slots are accessed. This field is ignored if the header is not present.		Value	Name	Description	00h	SG4x2	SIMD4x2	01h	SG8L	Use low 8 slots	02h	SG8U	Use high 8 slots	03h	Reserved	Ignored
Format:	Enumeration																				
Controls which 8 bits of Pixel/Sample Mask in the message header are ANDed with the execution mask to determine which slots are accessed. This field is ignored if the header is not present.																					
Value	Name	Description																			
00h	SG4x2	SIMD4x2																			
01h	SG8L	Use low 8 slots																			
02h	SG8U	Use high 8 slots																			
03h	Reserved	Ignored																			

Slot Group Select Render Cache Message Descriptor Control Field

MDC_RT_SGS - Slot Group Select Render Cache Message Descriptor Control Field											
Source:	BSpec										
Size (in bits):	1										
Default Value:	0x00000000										
DWord	Bit	Description									
0	0	<p>Slot Group Select</p> <p>This field selects whether slots 15:0 or slots 31:16 are used for bypassed data. Bypassed data includes the antialias alpha, multisample coverage mask, and if the header is not present also includes the X/Y addresses and pixel enables. For 8- and 16-pixel dispatches, SLOTGRP_LO must be selected on every message. For 32-pixel dispatches, this field must be set correctly for each message based on which slots are currently being processed.</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;">00h</td> <td style="text-align: center;">SLOTGRP_LO</td> <td>Choose bypassed data for slots 15:0</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">SLOTGRP_HI</td> <td>Choose bypassed data for slots 31:16</td> </tr> </tbody> </table>	Value	Name	Description	00h	SLOTGRP_LO	Choose bypassed data for slots 15:0	01h	SLOTGRP_HI	Choose bypassed data for slots 31:16
Value	Name	Description									
00h	SLOTGRP_LO	Choose bypassed data for slots 15:0									
01h	SLOTGRP_HI	Choose bypassed data for slots 31:16									

SO_DECL

SO_DECL												
Source:	RenderCS											
Size (in bits):	16											
Default Value:	0x00000000											
<p>A list of SO_DECL structures are passed in the 3DSTATE_SO_DECL_LIST command. Each structure specifies either (a) the source and destination of an up-to-4-DWord appending write into an SO buffer, or (b) how many DWords to skip over in the destination SO buffer (i.e., a "hole" where the previous buffer contents are maintained).</p>												
DWord	Bit	Description										
0	15:14	Reserved Format: _____ MBZ										
	13:12	Output Buffer Slot Format: _____ U2 Buffer Index This field selects the destination output buffer slot.										
	11	Hole Flag Format: _____ Flag If set, the Component Mask field indirectly specifies a number of 32-bit locations to skip over (leave unmodified in memory) in the selected output buffer. The Register Index field is ignored. The only permitted Component Mask values are as follows: <table border="1" style="margin-left: 20px;"> <tr><td>0x0</td><td>No Dwords are skipped over (SO_DECL performs no operation)</td></tr> <tr><td>0x1 (X)</td><td>Skip 1 DWord</td></tr> <tr><td>0x3 (XY)</td><td>Skip 2 DWords</td></tr> <tr><td>0x7 (XYZ)</td><td>Skip 3 DWords</td></tr> <tr><td>0xF (XYZW)</td><td>Skip 4 DWords</td></tr> </table>	0x0	No Dwords are skipped over (SO_DECL performs no operation)	0x1 (X)	Skip 1 DWord	0x3 (XY)	Skip 2 DWords	0x7 (XYZ)	Skip 3 DWords	0xF (XYZW)	Skip 4 DWords
	0x0	No Dwords are skipped over (SO_DECL performs no operation)										
	0x1 (X)	Skip 1 DWord										
0x3 (XY)	Skip 2 DWords											
0x7 (XYZ)	Skip 3 DWords											
0xF (XYZW)	Skip 4 DWords											
10	Reserved Format: _____ MBZ											
9:4	Register Index Format: _____ U6 128-bit granular offset into the source vertex read data If Hole Flag is clear, this field specifies the 128-bit offset into the source vertex data which supplies the source data to be written to the destination buffer, where the individual 32-component destination locations are selected by Component Mask. e.g., Register Index 0 corresponds with the first 128 bits of the data read from the vertex URB entry (as per corresponding Vertex Read Offset state) There is only enough internal storage for the 128-bit vertex header and 32 128-bit vertex attributes.											

SO_DECL															
	<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>[0,32]</td> <td></td> </tr> <tr> <td>0h</td> <td style="text-align: center;">[Default]</td> </tr> </tbody> </table>	Value	Name	[0,32]		0h	[Default]								
Value	Name														
[0,32]															
0h	[Default]														
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>It is the responsibility of software to map any API-visible source data specifications (e.g., vertex register number) into 128-bit granular URB read offsets.</td> </tr> </tbody> </table>	Programming Notes	It is the responsibility of software to map any API-visible source data specifications (e.g., vertex register number) into 128-bit granular URB read offsets.												
Programming Notes															
It is the responsibility of software to map any API-visible source data specifications (e.g., vertex register number) into 128-bit granular URB read offsets.															
3:0	<p>Component Mask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td>MASK 4-bit Mask</td> </tr> </table> <p>This field is a 4-bit bitmask that selects which contiguous 32-bit component(s) are either written or skipped-over in the destination buffer. If this field is zero the SO_DECL operation is effectively a no-op. No data will be appended to the destination and the destination buffer's write pointer will not be advanced. If the Hole Flag is set, this field (if non-zero) indirectly specifies how much the destination buffer's write pointer should be advanced. See Hole Flag description above for restrictions on this field. If the Hole Flag is clear, this field (if non-zero) selects which source components are to be written to the destination buffer. The components must be contiguous, e.g. YZW is legal, but XZW is not. The selected source components are written to the destination buffer starting at the current write pointer, and then the write pointer is advanced past the written data. E.g., if YZW is specified, the three (YZW) components of the source register will be written to the destination buffer at the current write pointer, and the write pointer will be advanced by 3 DWords.</p> <table border="1" style="width: 100%;"> <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 style="text-align: center;">[Default]</td> </tr> <tr> <td>xxx1b</td> <td>SO_DECL_COMPMASK_X</td> </tr> <tr> <td>xx1xb</td> <td>SO_DECL_COMPMASK_Y</td> </tr> <tr> <td>x1xxb</td> <td>SO_DECL_COMPMASK_Z</td> </tr> <tr> <td>1xxxb</td> <td>SO_DECL_COMPMASK_W</td> </tr> </tbody> </table>	Format:	MASK 4-bit Mask	Value	Name	0h	[Default]	xxx1b	SO_DECL_COMPMASK_X	xx1xb	SO_DECL_COMPMASK_Y	x1xxb	SO_DECL_COMPMASK_Z	1xxxb	SO_DECL_COMPMASK_W
Format:	MASK 4-bit Mask														
Value	Name														
0h	[Default]														
xxx1b	SO_DECL_COMPMASK_X														
xx1xb	SO_DECL_COMPMASK_Y														
x1xxb	SO_DECL_COMPMASK_Z														
1xxxb	SO_DECL_COMPMASK_W														

SO_DECL_ENTRY

SO_DECL_ENTRY				
Source:	RenderCS			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0..1	63:48	Stream 3 Decl <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">SO_DECL</td> </tr> </table> This field contains Stream 3 SO_DECL [n]	Format:	SO_DECL
	Format:	SO_DECL		
	47:32	Stream 2 Decl <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">SO_DECL</td> </tr> </table> This field contains Stream 2 SO_DECL [n]	Format:	SO_DECL
	Format:	SO_DECL		
31:16	Stream 1 Decl <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">SO_DECL</td> </tr> </table> This field contains Stream 1 SO_DECL [n]	Format:	SO_DECL	
Format:	SO_DECL			
15:0	Stream 0 Decl <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">SO_DECL</td> </tr> </table> This field contains Stream 0 SO_DECL [n]	Format:	SO_DECL	
Format:	SO_DECL			

Split_coding_unit_flags

Split_coding_unit_flags				
Source:	VideoCS			
Size (in bits):	21			
Default Value:	0x00000000			
Contains the split level flags, level 0 through 2.				
DWord	Bit	Description		
0	20	Split_flag_level0 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="width: 50px; text-align: center;">U1</td></tr></table>		U1
		U1		
	19:16	Split_flag_level1 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="width: 50px; text-align: center;">U4</td></tr></table> [19:16] is in raster order. Bit16 is for partition0 in raster order.		U4
		U4		
	15:12	Split_flag_level2 level1part3 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="width: 50px; text-align: center;">U4</td></tr></table> Split flags for bit19 partition. [15:12] is in raster order. Bit12 is for partition0 in raster order.		U4
		U4		
11:8	Split_flag_level2 level1part2 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="width: 50px; text-align: center;">U4</td></tr></table> Split flags for bit18 partition. [11:8] is in raster order. Bit8 is for partition0 in raster order.		U4	
	U4			
7:4	Split_flag_level2 level1part1 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="width: 50px; text-align: center;">U4</td></tr></table> Split flags for bit17 partition. [7:4] is in raster order. Bit4 is for partition0 in raster order.		U4	
	U4			
3:0	Split_flag_level2 level1part0 Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"></td><td style="width: 50px; text-align: center;">U4</td></tr></table> Split flags for bit16 partition. [3:0] is in raster order. Bit0 is for partition0 in raster order.		U4	
	U4			

SplitBaseAddress4KByteAligned

SplitBaseAddress4KByteAligned		
Source:	BSpec	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
Specifies a 64-bit (48-bit canonical) 4K-byte aligned memory base address.		
DWord	Bit	Description
0	31:12	Base Address Low Format: GraphicsAddress[31:12]
	11:0	Reserved Format: MBZ
1	31:16	Reserved Format: MBZ
	15:0	Base Address High Format: GraphicsAddress[47:32]

SplitBaseAddress64ByteAligned

SplitBaseAddress64ByteAligned		
Source:	BSpec	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
Specifies a 64-bit (48-bit canonical) 64-byte aligned memory base address.		
DWord	Bit	Description
0	31:6	Base Address Low Format: GraphicsAddress[31:6]
	5:0	Reserved Format: MBZ
1	31:16	Reserved Format: MBZ
	15:0	Base Address High Format: GraphicsAddress[47:32]

SrcRegNum

SrcRegNum											
Source:	Eulsa										
Size (in bits):	8										
Default Value:	0x00000000										
Description											
<p>Register Number The register number for the operand. For a GRF register, is the part of a register address that aligns to a 256-bit (32-byte) boundary. For an ARF register, this field is encoded such that MSBs identify the architecture register type and LSBs provide the register number. An ARF register can only be dst or src0. Any src1 or src2 operands cannot be ARF registers. RegNum and SubRegNum together provide the byte-aligned address for the origin of a register region. RegNum provides bits 12:5 of that address. For one-source and two-source instructions, SubregNum provides bits 4:0. For three-source instructions, the address must be DWord-aligned; SubRegNum provides bits 4:2 of the address and bits 1:0 are zero. This field is present for the direct addressing mode and not present for indirect addressing. This field applies to both source and destination operands.</p>											
DWord	Bit	Description									
0	7:0	<p>Source Register Number</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-127</td> <td>If {Dst/Src0/Src1/Src2}.RegFile==GRF</td> <td></td> </tr> <tr> <td>0-0ffh</td> <td>If {Dst/Src0/Src1/Src2}.RegFile==ARF</td> <td>This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.</td> </tr> </tbody> </table>	Value	Name	Description	0-127	If {Dst/Src0/Src1/Src2}.RegFile==GRF		0-0ffh	If {Dst/Src0/Src1/Src2}.RegFile==ARF	This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.
Value	Name	Description									
0-127	If {Dst/Src0/Src1/Src2}.RegFile==GRF										
0-0ffh	If {Dst/Src0/Src1/Src2}.RegFile==ARF	This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.									

SrcSubRegNum

SrcSubRegNum											
Source:	Eulsa										
Size (in bits):	5										
Default Value:	0x00000000										
Description											
<p>Subregister Number The subregister number for the operand. For a GRF register, is the byte address within a 256-bit (32-byte) register. For an ARF register, determines the sub-register number according to the specified encoding for the given architecture register. RegNum and SubRegNum together provide the byte-aligned address for the origin of a GRF register region. RegNum provides bits 12:5 of that address. For one-source and two-source instructions, SubregNum provides bits 4:0. For three-source instructions, the address must be DWord-aligned; SubRegNum provides bits 4:2 of the address and bits 1:0 are zero.</p>											
Programming Notes											
<p>Note: The recommended instruction syntax uses subregister numbers within the GRF in units of actual data element size, corresponding to the data type used. For example for the F (Float) type, the assembler syntax uses subregister numbers 0 to 7, corresponding to subregister byte addresses of 0 to 28 in steps of 4, the element size.</p>											
DWord	Bit	Description									
0	4:0	<p>Source Sub Register Number</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-31</td> <td>If {Dst/Src0/Src1/Src2}.RegFile==GRF</td> <td></td> </tr> <tr> <td>0-Offh</td> <td>If {Dst/Src0/Src1/Src2}.RegFile==ARF</td> <td>This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.</td> </tr> </tbody> </table>	Value	Name	Description	0-31	If {Dst/Src0/Src1/Src2}.RegFile==GRF		0-Offh	If {Dst/Src0/Src1/Src2}.RegFile==ARF	This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.
Value	Name	Description									
0-31	If {Dst/Src0/Src1/Src2}.RegFile==GRF										
0-Offh	If {Dst/Src0/Src1/Src2}.RegFile==ARF	This field is used to encode the architecture register as well as providing the register number. See GEN Execution Environment chapter for details.									

SRD Interrupt Bit Definition

SRD Interrupt Bit Definition		
Source:	BSpec	
Size (in bits):	32	
Default Value:	0x00000000	
The SRD Interrupt Registers all share the same bit definitions from this table.		
DWord	Bit	Description
0	31:30	Reserved
	29:27	Reserved
	26	Reserved
	25	SRD_Exit_C This event occurs on the first blank start after SRD exit on transcoder C.
	24	SRD_PreWarn_C This event occurs two display frames prior to entering SRD on transcoder C.
	23:19	Reserved
	18	Reserved
	17	SRD_Exit_B This event occurs on the first blank start after SRD exit on transcoder B.
	16	SRD_PreWarn_B This event occurs two display frames prior to entering SRD on transcoder B.
	15:11	Reserved
	10	Reserved
	9	SRD_Exit_A This event occurs on the first blank start after SRD exit on transcoder A.
	8	SRD_PreWarn_A This event occurs two display frames prior to entering SRD on transcoder A.
	7:3	Reserved
	2	SRD_Aux_Error_EDP This event occurs on the rising edge of the SRD Aux error (receive error or timeout) indication.
	1	SRD_Exit_EDP This event occurs on the first blank start after SRD exit on transcoder EDP.
0	SRD_PreWarn_EDP This event occurs two display frames prior to entering SRD on transcoder EDP.	

Stateless Binding Table Index Message Descriptor Control Field

MDC_STATELESS - Stateless Binding Table Index Message Descriptor Control Field																		
Source:	BSpec																	
Size (in bits):	8																	
Default Value:	0x000000FF																	
DWord	Bit	Description																
0	7:0	<p>Binding Table Index</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enumeration</td> </tr> <tr> <td colspan="2">Specifies the message is Stateless</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">0FFh</td> <td>A32_A64 [Default]</td> <td>Specifies a A32 or A64 Stateless access that is locally coherent (coherent within a thread group)</td> </tr> <tr> <td style="text-align: center;">0FDh</td> <td>A32_A64_NC</td> <td>Specifies a A32 or A64 Stateless access that is non-coherent (coherent within a thread).</td> </tr> <tr> <td style="text-align: center;">Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </table> <p style="text-align: center;">Restriction</p> <p>When using A32_A64_NC, SW must ensure that 2 threads do not both access the same cache line (64B)</p>	Format:	Enumeration	Specifies the message is Stateless		Value	Name	Description	0FFh	A32_A64 [Default]	Specifies a A32 or A64 Stateless access that is locally coherent (coherent within a thread group)	0FDh	A32_A64_NC	Specifies a A32 or A64 Stateless access that is non-coherent (coherent within a thread).	Others	Reserved	Ignored
Format:	Enumeration																	
Specifies the message is Stateless																		
Value	Name	Description																
0FFh	A32_A64 [Default]	Specifies a A32 or A64 Stateless access that is locally coherent (coherent within a thread group)																
0FDh	A32_A64_NC	Specifies a A32 or A64 Stateless access that is non-coherent (coherent within a thread).																
Others	Reserved	Ignored																

Stateless Block Message Header

MH_A32_GO - Stateless Block Message Header				
Source:	DataPort 0			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0-1	63:0	Reserved		
		Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">Ignore</td></tr></table> Ignored		Ignore
	Ignore			
2	31:0	Global Offset		
		Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U32</td></tr></table> Specifies the global element index into the buffer, in units of Owords, Dwords, or Bytes (depending on the message).		U32
			U32	
Programming Notes				
If the address offset calculated with the Buffer Base Address and Global Offset is greater than the PTSS size or the GeneralStateBufferSize, then the access is Out-of-Bounds.				
3	31:0	Per Thread Scratch Space		
		Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">MHC_PTSS</td></tr></table> Specifies amount of scratch space used by this thread, for Stateless bounds checking.		MHC_PTSS
	MHC_PTSS			
4	31:0	Reserved		
		Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">Ignore</td></tr></table> Ignored		Ignore
	Ignore			
5	31:0	Buffer Base Address		
		Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">MHC_A32_BBA</td></tr></table>		MHC_A32_BBA
			MHC_A32_BBA	
		Description		
Specifies the surface address offset page [31:10] for A32 stateless messages. 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 $\geq 2^{48}$.				
6-7	63:0	Reserved		
		Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">Ignore</td></tr></table> Ignored		Ignore
	Ignore			

Stateless Scaled Data Port 2 Message Header

MH2_A32 - Stateless Scaled Data Port 2 Message Header				
Source:	DataPort 2			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	<p>Surface Pitch</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">MHC_PITCH</td> </tr> </table> <p>Specifies the 16-bit surface pitch to use with scaled A32 stateless messages.</p>	Format:	MHC_PITCH
Format:	MHC_PITCH			
1-4	127:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
5	31:0	<p>Buffer Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td style="text-align: center;">GeneralStateOffset[31:0]</td> </tr> </table> <p>Specifies the surface address offset page [31:0] for scaled A32 stateless messages. The address must be Dword aligned (bits [1:0] = 0).</p>	Format:	GeneralStateOffset[31:0]
Format:	GeneralStateOffset[31:0]			
6-7	63:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			

Stateless Scaled Data Port 2 Pixel Mask Message Header

MH2_A32_PSM - Stateless Scaled Data Port 2 Pixel Mask Message Header				
Source:	DataPort 2			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x0000FFFF			
DWord	Bit	Description		
0	31:0	<p>Surface Pitch</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MHC_PITCH</td> </tr> </table> <p>Specifies the 16-bit surface pitch to use with scaled A32 stateless messages.</p>	Format:	MHC_PITCH
Format:	MHC_PITCH			
1-4	127:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
5	31:0	<p>Buffer Base Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">GeneralStateOffset[31:0]</td> </tr> </table> <p>Specifies the surface address offset page [31:0] for scaled A32 stateless messages. The address must be Dword aligned (bits [1:0] = 0).</p>	Format:	GeneralStateOffset[31:0]
Format:	GeneralStateOffset[31:0]			
6	31:0	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
7	31:0	<p>Pixel Sample Mask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MHC_PSM</td> </tr> </table> <p>Specifies the 16-bit Pixel/Sample Mask used with SIMD16 and SIMD8 untyped surfaces.</p>	Format:	MHC_PSM
Format:	MHC_PSM			

Stateless Surface Message Header

MH1_A32 - Stateless Surface Message Header				
Source:	DataPort 1			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0-4	159:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Ignore</td> </tr> </table> Ignored	Format:	Ignore
Format:	Ignore			
5	31:0	Buffer Base Address <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="color: red;">MHC_A32_BBA</td> </tr> </table> Specifies the surface address offset page [31:10] for A32 stateless messages.	Format:	MHC_A32_BBA
Format:	MHC_A32_BBA			
6-7	63:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Ignore</td> </tr> </table> Ignored	Format:	Ignore
Format:	Ignore			

Stateless Surface Pixel Mask Message Header

MH1_A32_PSM - Stateless Surface Pixel Mask Message Header				
Source:	DataPort 1			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x0000FFFF			
DWord	Bit	Description		
0-4	159:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
5	31:0	<p>Buffer Base Address</p> <table border="1"> <tr> <td>Format:</td> <td>MHC_A32_BBA</td> </tr> </table> <p>Specifies the surface address offset page [31:10] for A32 stateless messages.</p>	Format:	MHC_A32_BBA
Format:	MHC_A32_BBA			
6	31:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
Format:	Ignore			
7	31:0	<p>Pixel Sample Mask</p> <table border="1"> <tr> <td>Format:</td> <td>MHC_PSM</td> </tr> </table> <p>Specifies the 16-bit Pixel/Sample Mask used with SIMD16 and SIMD8 surfaces.</p>	Format:	MHC_PSM
Format:	MHC_PSM			

Stencil Message Data Payload Register

MDPR_STENCIL - Stencil Message Data Payload Register				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:24	Stencil3 <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Stencil for Slot 3.	Format:	U8
	Format:	U8		
	23:16	Stencil2 <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Stencil for Slot 2.	Format:	U8
	Format:	U8		
15:8	Stencil1 <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Stencil for Slot 1.	Format:	U8	
Format:	U8			
7:0	Stencil0 <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Stencil for Slot 0.	Format:	U8	
Format:	U8			
1	31:24	Stencil7 <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Stencil for Slot 7.	Format:	U8
	Format:	U8		
	23:16	Stencil6 <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Stencil for Slot 6.	Format:	U8
	Format:	U8		
15:8	Stencil5 <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Stencil for Slot 5.	Format:	U8	
Format:	U8			
7:0	Stencil4 <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Stencil for Slot 4.	Format:	U8	
Format:	U8			

MDPR_STENCIL - Stencil Message Data Payload Register						
2-7	191:0	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Ignore</td> </tr> <tr> <td colspan="2">Ignored</td> </tr> </table>	Format:	Ignore	Ignored	
Format:	Ignore					
Ignored						

Subset Atomic Integer Ternary Operation Message Descriptor Control Field

MDC_AOP3S - Subset Atomic Integer Ternary Operation Message Descriptor Control Field																	
Source:	BSpec																
Size (in bits):	4																
Default Value:	0x0000000E																
DWord	Bit	Description															
0	3:0	<p>Atomic Integer Operation Type</p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">Enumeration</td> </tr> <tr> <td colspan="3">Specifies the atomic integer ternary operation to be performed</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>0Eh</td> <td>AOP_CMPWR [Default]</td> <td>$\text{new_dst} = (\text{src0} == \text{old_dst}) ? \text{src1} : \text{old_dst}$</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </table> <p style="text-align: center;">Programming Notes</p> <p>When Return Data Control is set, old_dst is returned.</p>	Format:	Enumeration		Specifies the atomic integer ternary operation to be performed			Value	Name	Description	0Eh	AOP_CMPWR [Default]	$\text{new_dst} = (\text{src0} == \text{old_dst}) ? \text{src1} : \text{old_dst}$	Others	Reserved	Ignored
Format:	Enumeration																
Specifies the atomic integer ternary operation to be performed																	
Value	Name	Description															
0Eh	AOP_CMPWR [Default]	$\text{new_dst} = (\text{src0} == \text{old_dst}) ? \text{src1} : \text{old_dst}$															
Others	Reserved	Ignored															

Subset Reversed SIMD Mode 2 Message Descriptor Control Field

MDC_SM2RS - Subset Reversed SIMD Mode 2 Message Descriptor Control Field											
Source:	BSpec										
Size (in bits):	1										
Default Value:	0x00000001										
DWord	Bit	Description									
0	0	SIMD Mode Format: Enumeration Specifies the SIMD mode of the message (number of slots processed) <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> <td>Not used</td> </tr> <tr> <td>01h</td> <td>SIMD8 [Default]</td> <td>SIMD8</td> </tr> </tbody> </table>	Value	Name	Description	0h	Reserved	Not used	01h	SIMD8 [Default]	SIMD8
Value	Name	Description									
0h	Reserved	Not used									
01h	SIMD8 [Default]	SIMD8									

Subset SIMD Mode 2 Message Descriptor Control Field

MDC_SM2S - Subset SIMD Mode 2 Message Descriptor Control Field											
Source:	BSpec										
Size (in bits):	1										
Default Value:	0x00000000										
DWord	Bit	Description									
0	0	SIMD Mode Format: Enumeration Specifies the SIMD mode of the message (number of slots processed)									
		<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;">00h</td> <td>SIMD8</td> <td>SIMD8</td> </tr> <tr> <td style="text-align: center;">01h</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Value	Name	Description	00h	SIMD8	SIMD8	01h	Reserved	Ignored
Value	Name	Description									
00h	SIMD8	SIMD8									
01h	Reserved	Ignored									

Subset SIMD Mode 3 Message Descriptor Control Field

MDC_SM3S - Subset SIMD Mode 3 Message Descriptor Control Field																					
Source:	BSpec																				
Size (in bits):	2																				
Default Value:	0x00000000																				
DWord	Bit	Description																			
0	1:0	<p>SIMD Mode</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> <tr> <td colspan="2">Specifies the SIMD mode of the message (number of slots processed)</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">SIMD4x2</td> <td style="text-align: center;">SIMD4x2</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Ignored</td> </tr> <tr> <td style="text-align: center;">02h</td> <td style="text-align: center;">SIMD8</td> <td style="text-align: center;">SIMD8</td> </tr> <tr> <td style="text-align: center;">03h</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Ignored</td> </tr> </table>	Format:	Enumeration	Specifies the SIMD mode of the message (number of slots processed)		Value	Name	Description	00h	SIMD4x2	SIMD4x2	01h	Reserved	Ignored	02h	SIMD8	SIMD8	03h	Reserved	Ignored
Format:	Enumeration																				
Specifies the SIMD mode of the message (number of slots processed)																					
Value	Name	Description																			
00h	SIMD4x2	SIMD4x2																			
01h	Reserved	Ignored																			
02h	SIMD8	SIMD8																			
03h	Reserved	Ignored																			

Subspan Render Target Message Header Control

MHC_RT_SUBSPAN - Subspan Render Target Message Header Control				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:16	Y <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">U16</td> </tr> </table> Y coordinate for upper-left pixel of this subspan	Format:	U16
	Format:	U16		
15:0	X <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">U16</td> </tr> </table> X coordinate for upper-left pixel of this subspan	Format:	U16	
Format:	U16			

Surface Binding Table Index Message Descriptor Control Field

MDC_BTS - Surface Binding Table Index Message Descriptor Control Field																	
Source:	BSpec																
Size (in bits):	8																
Default Value:	0x00000000																
DWord	Bit	Description															
0	7:0	Binding Table Index Format: Enumeration Specifies the Binding Table index for the message, which must be a Surface State Model.															
		<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>00h-0EFh</td> <td>BTS</td> <td>Index of Binding Table State Surfaces</td> </tr> <tr> <td>F0h-0FBh</td> <td>Reserved</td> <td>Reserved for future use</td> </tr> <tr> <td>0FCh</td> <td>SSO</td> <td>Specifies a Surface State Offset supplied by the extended message descriptor</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Value	Name	Description	00h-0EFh	BTS	Index of Binding Table State Surfaces	F0h-0FBh	Reserved	Reserved for future use	0FCh	SSO	Specifies a Surface State Offset supplied by the extended message descriptor	Others	Reserved	Ignored
Value	Name	Description															
00h-0EFh	BTS	Index of Binding Table State Surfaces															
F0h-0FBh	Reserved	Reserved for future use															
0FCh	SSO	Specifies a Surface State Offset supplied by the extended message descriptor															
Others	Reserved	Ignored															

Surface or Stateless Binding Table Index Message Descriptor Control Field

MDC_BTS_A32 - Surface or Stateless Binding Table Index Message Descriptor Control Field																							
Source:	BSpec																						
Size (in bits):	8																						
Default Value:	0x00000000																						
DWord	Bit	Description																					
0	7:0	Binding Table Index																					
		Format: Enumeration																					
		Specifies the surface for the message, either Surface State Model or Stateless.																					
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h-0EFh</td> <td>BTS</td> <td>Index of Binding Table State Surfaces</td> </tr> <tr> <td>F0h-0FBh</td> <td>Reserved</td> <td>Reserved for future use</td> </tr> <tr> <td>0FCh</td> <td>SSO</td> <td>Specifies a Surface State Offset supplied by the extended message descriptor</td> </tr> <tr> <td>0FFh</td> <td>A32_A64</td> <td>Specifies a A32 or A64 Stateless access that is locally coherent (coherent within a thread group)</td> </tr> <tr> <td>0FDh</td> <td>A32_A64_NC</td> <td>Specifies a A32 or A64 Stateless access that is non-coherent (coherent within a thread).</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Value	Name	Description	00h-0EFh	BTS	Index of Binding Table State Surfaces	F0h-0FBh	Reserved	Reserved for future use	0FCh	SSO	Specifies a Surface State Offset supplied by the extended message descriptor	0FFh	A32_A64	Specifies a A32 or A64 Stateless access that is locally coherent (coherent within a thread group)	0FDh	A32_A64_NC	Specifies a A32 or A64 Stateless access that is non-coherent (coherent within a thread).	Others	Reserved	Ignored
		Value	Name	Description																			
		00h-0EFh	BTS	Index of Binding Table State Surfaces																			
		F0h-0FBh	Reserved	Reserved for future use																			
		0FCh	SSO	Specifies a Surface State Offset supplied by the extended message descriptor																			
		0FFh	A32_A64	Specifies a A32 or A64 Stateless access that is locally coherent (coherent within a thread group)																			
		0FDh	A32_A64_NC	Specifies a A32 or A64 Stateless access that is non-coherent (coherent within a thread).																			
Others	Reserved	Ignored																					
Restriction																							
When using A32_A64_NC, SW must ensure that 2 threads do not both access the same cache line (64B)																							

Surface Pitch Message Header Control

MHC_PITCH - Surface Pitch Message Header Control				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:16	<p>Reserved</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
	Format:	Ignore		
15:0	<p>Surface Pitch</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>U16</td> </tr> </table> <p>Specifies the surface width (pitch) to be used for Stateless messages. The Surface Pitch is specified as (#items - 1). The items are specified in the message, and can be bytes, Dwords, or Qwords. The target element address offset is:</p> $\text{BufferBaseAddress} + U * (\text{SurfacePitch} + 1)$ <p>where U is the 32-bit element offset delivered in message address payload.</p>	Format:	U16	
Format:	U16			

Surface Pixel Mask Message Header

MH1_BTS_PSM - Surface Pixel Mask Message Header		
Source:	DataPort 1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x0000FFFF	
DWord	Bit	Description
0-6	223:0	Reserved
		Format: Ignore Ignored
7	31:0	Pixel Sample Mask
		Format: MHC_PSM Specifies the 16-bit Pixel/Sample Mask used with SIMD16 and SIMD8 surfaces.

SW Generated BINDING_TABLE_STATE

SW Generated BINDING_TABLE_STATE				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
Description				
<p>The binding table binds surfaces to logical resource indices used by shaders and other compute engine kernels. It is stored as an array of up to 256 elements, each of which contains one dword as defined here. The start of each element is spaced one dword apart.</p> <p>The first element of the binding table is aligned to a 64-byte boundary.</p> <p>Binding table indexes beyond 256 will automatically be mapped to entry 0 by the HW, w/ the exception of any messages which support the special indexes 240 through 255, inclusive.</p>				
DWord	Bit	Description		
0	31:6	<p>Surface State Pointer</p> <table border="1"> <tr> <td>Format:</td> <td>SurfaceStateOffset[31:6]</td> </tr> </table> <p>This 64-byte aligned address points to a surface state block. This pointer is relative to the Surface State Base Address</p>	Format:	SurfaceStateOffset[31:6]
	Format:	SurfaceStateOffset[31:6]		
	5	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
4:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ	
Format:	MBZ			

MDP_RTW_ZMA8 - SZ OM S0A SIMD8 Render Target Data Payload

6.0-6.7	255:0	Source Depth	
		Format:	MDP_DW_SIMD8
Slots [7:0] Source Depth			

MDP_RTW_ZMA16 - SZ OM S0A SIMD16 Render Target Data Payload		
7.0-8.7	511:0	Blue
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD16</td> </tr> </table> Slots [15:0] Blue
Format:	MDP_DW_SIMD16	
9.0-10.7	511:0	Alpha
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD16</td> </tr> </table> Slots [15:0] Alpha
Format:	MDP_DW_SIMD16	
11.0-12.7	511:0	Source Depth
		<table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD16</td> </tr> </table> Slots [15:0] Source Depth
Format:	MDP_DW_SIMD16	

MDP_RTW_ZM8DS - SZ OM SIMD8 Dual Source Render Target Data Payload		
5.0-5.7	255:0	Src1 Red Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Red
6.0-6.7	255:0	Src1 Green Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Green
7.0-7.7	255:0	Src1 Blue Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Blue
8.0-8.7	255:0	Src1 Alpha Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Alpha
9.0-9.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] or [15:8] of Source Depth

SZ OM SIMD8 Render Target Data Payload

MDP_RTW_ZM8 - SZ OM SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	oMask Format: MDPR_OMASK Slots [7:0] oMask. Upper half ignored.
1.0-1.7	255:0	Red Format: MDP_DW_SIMD8 Slots [7:0] Red
2.0-2.7	255:0	Green Format: MDP_DW_SIMD8 Slots [7:0] Green
3.0-3.7	255:0	Blue Format: MDP_DW_SIMD8 Slots [7:0] Blue
4.0-4.7	255:0	Alpha Format: MDP_DW_SIMD8 Slots [7:0] Alpha
5.0-5.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] Source Depth

MDP_RTW_ZM16 - SZ OM SIMD16 Render Target Data Payload				
5.0-5.7	255:0	Blue[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Blue	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
6.0-6.7	255:0	Blue[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Blue	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
7.0-7.7	255:0	Alpha[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Alpha	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
8.0-8.7	255:0	Alpha[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Alpha	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
9.0-9.7	255:0	Source Depth[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Source Depth	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
10.0-10.7	255:0	Source Depth[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Source Depth	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			

SZ S0A SIMD8 Render Target Data Payload

MDP_RTW_ZA8 - SZ S0A SIMD8 Render Target Data Payload		
Source:	BSpec	
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	Source 0 Alpha Format: MDP_DW_SIMD8 Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	Red Format: MDP_DW_SIMD8 Slots [7:0] Red
2.0-2.7	255:0	Green Format: MDP_DW_SIMD8 Slots [7:0] Green
3.0-3.7	255:0	Blue Format: MDP_DW_SIMD8 Slots [7:0] Blue
4.0-4.7	255:0	Alpha Format: MDP_DW_SIMD8 Slots [7:0] Alpha
5.0-5.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] Source Depth

MDP_RTW_ZA16 - SZ S0A SIMD16 Render Target Data Payload

5.0-5.7	255:0	Green[15:8]	
		Format:	MDP_DW_SIMD8
Slots [15:8] Green			
6.0-6.7	255:0	Blue[7:0]	
		Format:	MDP_DW_SIMD8
Slots [7:0] Blue			
7.0-7.7	255:0	Blue[15:7]	
		Format:	MDP_DW_SIMD8
Slots [15:8] Blue			
8.0-8.7	255:0	Alpha[7:0]	
		Format:	MDP_DW_SIMD8
Slots [7:0] Alpha			
9.0-9.7	255:0	Alpha[15:8]	
		Format:	MDP_DW_SIMD8
Slots [15:8] Alpha			
10.0-10.7	255:0	Source Depth[7:0]	
		Format:	MDP_DW_SIMD8
Slots [7:0] Source Depth			
11.0-11.7	255:0	Source Depth[15:8]	
		Format:	MDP_DW_SIMD8
Slots [15:8] Source Depth			

MDP_RTW_Z8DS - SZ SIMD8 Dual Source Render Target Data Payload		
5.0-5.7	255:0	Src1 Green Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Green
6.0-6.7	255:0	Src1 Blue Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Blue
7.0-7.7	255:0	Src1 Alpha Format: MDP_DW_SIMD8 Slots[7:0] or [15:8] of Src1 Alpha
8.0-8.7	255:0	Source Depth Format: MDP_DW_SIMD8 Slots [7:0] or [15:8] of Source Depth

MDP_RTW_Z16 - SZ SIMD16 Render Target Data Payload				
5.0-5.7	255:0	Blue[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Blue	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
6.0-6.7	255:0	Alpha[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Alpha	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
7.0-7.7	255:0	Alpha[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Alpha	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
8.0-8.7	255:0	Source Depth[7:0] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [7:0] Source Depth	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			
9.0-9.7	255:0	Source Depth[15:8] <table border="1"> <tr> <td>Format:</td> <td>MDP_DW_SIMD8</td> </tr> </table> Slots [15:8] Source Depth	Format:	MDP_DW_SIMD8
Format:	MDP_DW_SIMD8			

Thread EOT Message Descriptor

TS_EOT - Thread EOT Message Descriptor		
Source:	RenderCS	
Size (in bits):	32	
Default Value:	0x02000000	
<p>End of Thread message is sent to SFID_TS (07h) to end GPGPU and Media threads. The EU send instruction must also set the EOT control (bit 5) of the extended message descriptor.</p> <p>This message is sent with single register message payload, which is a copy of the R0 thread payload sent with the thread dispatch.</p>		
DWord	Bit	Description
0	31:29	Reserved Format: MBZ
	28:25	Message Length Default Value: 1h One GRF Format: U4
	24:20	Response Length Default Value: 0h Zero GRF Format: U5
	19	Header Present Format: MBZ
	18:1	Reserved Format: MBZ
	0	Message Type Default Value: 0h End Thread Format: Opcode End of Thread message opcode

Thread Spawn Message Descriptor

Thread Spawn Message Descriptor				
Source:	RenderCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:20	Reserved		
		Format:	MBZ	
	19	Header Present		
		Format:	MBZ	
	Programming Notes		This bit MBZ for all Thread Spawner messages.	
	18:5	Reserved		
		Format:	MBZ	
	4	Resource Select		
		This field specifies the resource associated with the action taken by the Opcode.		
		Value	Name	Description
	0	Spawn Child	Spawn a Child Thread	[Opcode] == 'Spawn Thread'
	1	Spawn Root	Spawn a Root Thread	[Opcode] == 'Spawn Thread'
	0	Dereference Resource	The URB Handle is Dereferenced	[Opcode] == 'Dereference Resource'
	1	Keep Resource	The URBHhandle is NOT Dereferenced	[Opcode] == 'Dereference Resource'
3:2	Reserved			
	Format:	MBZ		
1	Requester Type			
	This field indicates whether the requesting thread is a root thread or a child thread. If it is a root thread, when Opcode is 0, FF managed resources are dereferenced. If it is a child thread and Opcode is 0, no resource is dereferenced; no action is required by the TS.			
	Value	Name		
	0	Root Thread		
	1	Child Thread		

Thread Spawn Message Descriptor

	0	<p>Opcode</p> <p>Indicates the operation performed by the message. A root thread must terminate with a message to TS (Opcode == 0 and EOT == 1). A child thread should also terminate with such a message. A thread cannot terminate with an Opcode of "spawn thread".</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 style="text-align: center;">0</td> <td>Dereference Resource</td> <td>also used for end of thread</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Spawn Thread</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0	Dereference Resource	also used for end of thread	1	Spawn Thread	
Value	Name	Description									
0	Dereference Resource	also used for end of thread									
1	Spawn Thread										



TileW SIMD8 Data Control Dword

MDCD_TILEW - TileW SIMD8 Data Control Dword				
Source:	BSpec			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:8	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>Ignore</td> </tr> </table> <p>Ignored</p>	Format:	Ignore
	Format:	Ignore		
7:0	<p>Red</p> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Specifies the value of the red channel to be read or written.</p>	Format:	U8	
Format:	U8			

TileW SIMD8 Data Payload

MDP_TILEW_SIMD8 - TileW SIMD8 Data Payload				
Source:	BSpec			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0	31:0	Red Slot0 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDCD_TileW</td> </tr> </table> Specifies the Slot 0 red channel data	Format:	MDCD_TileW
Format:	MDCD_TileW			
0.1	31:0	Red Slot1 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDCD_TileW</td> </tr> </table> Specifies the Slot 1 red channel data	Format:	MDCD_TileW
Format:	MDCD_TileW			
0.2	31:0	Red Slot2 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDCD_TileW</td> </tr> </table> Specifies the Slot 2 red channel data	Format:	MDCD_TileW
Format:	MDCD_TileW			
0.3	31:0	Red Slot3 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDCD_TileW</td> </tr> </table> Specifies the Slot 3 red channel data	Format:	MDCD_TileW
Format:	MDCD_TileW			
0.4	31:0	Red Slot4 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDCD_TileW</td> </tr> </table> Specifies the Slot 4 red channel data	Format:	MDCD_TileW
Format:	MDCD_TileW			
0.5	31:0	Red Slot5 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDCD_TileW</td> </tr> </table> Specifies the Slot 5 red channel data	Format:	MDCD_TileW
Format:	MDCD_TileW			
0.6	31:0	Red Slot6 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDCD_TileW</td> </tr> </table> Specifies the Slot 6 red channel data	Format:	MDCD_TileW
Format:	MDCD_TileW			
0.7	31:0	Red Slot7 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;">MDCD_TileW</td> </tr> </table> Specifies the Slot 7 red channel data	Format:	MDCD_TileW
Format:	MDCD_TileW			

Transpose Message Header

MH_T - Transpose Message Header		
Source:	DataPort 1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	X Offset
		Format: S31
		X offset (in bytes) of the upper left corner of the block into the surface.
		Programming Notes
		This field must be a multiple of the Block Width in bytes. Must be DWORD aligned.
1	31:0	Y Offset
		Format: S31
		Y offset (in rows) of the upper left corner of the block into the surface.
		Programming Notes
		This field must be a multiple of the Block Height.
2	31:0	Block Dimensions
		Format: MHC_BDIM
		The height and width of the block to transpose.
3-7	159:0	Reserved
		Format: Ignore
		Ignored

Untyped Write Channel Mask Message Descriptor Control Field

MDC_UW_CMASK - Untyped Write Channel Mask Message Descriptor Control Field																						
Source:	BSpec																					
Size (in bits):	4																					
Default Value:	0x00000000																					
DWord	Bit	Description																				
0	3:0	<p>Mask</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enumeration</td> </tr> </table> <p>For untyped surface write messages, indicates which channels are included in the message payload and written to the 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> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">RGBA [Default]</td> <td>Red, Green, Blue, and Alpha are included</td> </tr> <tr> <td style="text-align: center;">08h</td> <td style="text-align: center;">RGB</td> <td>Red, Green, and Blue are included</td> </tr> <tr> <td style="text-align: center;">0Ch</td> <td style="text-align: center;">RG</td> <td>Red and Green are included</td> </tr> <tr> <td style="text-align: center;">0Eh</td> <td style="text-align: center;">R</td> <td>Red is included</td> </tr> <tr> <td style="text-align: center;">Others</td> <td style="text-align: center;">Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Format:	Enumeration	Value	Name	Description	00h	RGBA [Default]	Red, Green, Blue, and Alpha are included	08h	RGB	Red, Green, and Blue are included	0Ch	RG	Red and Green are included	0Eh	R	Red is included	Others	Reserved	Ignored
Format:	Enumeration																					
Value	Name	Description																				
00h	RGBA [Default]	Red, Green, Blue, and Alpha are included																				
08h	RGB	Red, Green, and Blue are included																				
0Ch	RG	Red and Green are included																				
0Eh	R	Red is included																				
Others	Reserved	Ignored																				

Upper Oword Block Data Payload

MDP_OW1U - Upper Oword Block Data Payload		
Source:	BSpec	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.3	127:0	Reserved
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Ignore</td> </tr> </table> Ignored
Format:	Ignore	
0.4-0.7	127:0	Oword
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U128</td> </tr> </table> Specifies the upper Oword data element
Format:	U128	

VC1

VC1				
Source:	VideoCS			
Size (in bits):	16			
Default Value:	0x00000000			
DWord	Bit	Description		
0	15:8	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		
	7	Syncmarker Error This flag indicates missing sync marker SEs coded in the bit-stream.		
	6	Mbmode SE Error This flag indicates inconsistent Macroblock SEs coded in the bit-stream.		
	5	Transformtype SE Error This flag indicates inconsistent transform type SEs coded in the bit-stream.		
	4	Coefficient Error This flag indicates inconsistent Coefficient SEs coded in the bit-stream.		
	3	Motion Vector SE Error This flag indicates inconsistent Motion Vector SEs coded in the bit-stream.		
	2	Coded Block Pattern CY SE Error This flag indicates inconsistent CBPCY SEs coded in the bit-stream.		
	1	Mquant Error This flag indicates inconsistent MQANT SEs coded in the bit-stream.		
0	MB Concealment Flag . Each pulse from this flag indicates one MB is concealed by hardware.			

VCS Hardware-Detected Error Bit Definitions

VCS Hardware-Detected Error Bit Definitions							
Source:	VideoCS						
Size (in bits):	16						
Default Value:	0x00000000						
DWord	Bit	Description					
0	15:3	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ			
		MBZ					
	2	Command Privilege Violation Error This bit is set if a command classified as privileged is parsed in a non-privileged batch buffer. The command will be converted to a NOOP and parsing will continue.					
	1	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ			
	MBZ						
0	Instruction Error This bit is set when the Renderer Instruction Parser detects an error while parsing an instruction. Instruction errors include: <ul style="list-style-type: none"> Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported). Defeated MI Instruction Opcodes: <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>1</td> <td></td> <td>Instruction Error detected</td> </tr> </tbody> </table>	Value	Name	Description	1		Instruction Error detected
Value	Name	Description					
1		Instruction Error detected					
Programming Notes							
This error indications cannot be cleared except by reset (i.e., it is a fatal error).							

VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS

VEB_DI_IECP_COMMAND_SURFACE_CONTROL_BITS																	
Source:	VideoEnhancementCS																
Size (in bits):	11																
Default Value:	0x00000000																
DWord	Bit	Description															
0	10:9	Tiled Resource Mode for Output Frame Surface Base Address 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															
	8	Memory Compression Mode Distinguishes Vertical from Horizontal compression.															
		<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																
7	Memory Compression Enable Format: <table border="1"><tr><td> </td><td>Enable</td></tr></table> Memory compression will be attempted for this surface.		Enable														
	Enable																
6:1	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																

VEBOX_ACE_LACE_STATE

VEBOX_ACE_LACE_STATE			
Source:	VideoEnhancementCS		
Size (in bits):	416		
Default Value:	0x00000068, 0x4C382410, 0x9C887460, 0xEBD8C4B0, 0x604C3824, 0xB09C8874, 0x0000D8C4, 0x04000400, 0x04000400, 0x04000400, 0x04000400, 0x04000400, 0x00000400		
This state structure contains the IECP State Table Contents for ACE state.			
DWord	Bit	Description	
0	31:16	Min_ACE_luma	
		Format: U16	
	15:14	LACE Single Histogram Set	
		This bit tells LACE which frames will be included in the histogram when the Deinterlacer is enabled.	
		Value	Name
		00b	Current
		01b	Previous
		10b	Current + Previous
	11b	Previous + Current	
	Programming Notes		
When the Deinterlacer is disabled, this field must be 00b. If DI Output Frames is set to only output a single field then the histogram can not be collected on the disabled field. This Field must be set to 00b when DN/DI First Frame is set to 1			
13	LACE Histogram Size		
	Value	Name	
	0	128-bin histogram	
1	256-bin histogram		
12	LACE Histogram Enable		
	Default Value: 0		
This bit enables the collection of LACE histogram data. If this bit is 0 then only the ACE histogram will be collected.			
11:7	Reserved		
	Format: MBZ		

VEBOX_ACE_LACE_STATE								
	6:2	Skin Threshold						
	Format: U5							
	Used for Y analysis (min/max) for pixels which are higher than skin threshold.							
		<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>[1,31]</td> <td></td> </tr> <tr> <td>26</td> <td style="text-align: center;">[Default]</td> </tr> </tbody> </table>	Value	Name	[1,31]		26	[Default]
Value	Name							
[1,31]								
26	[Default]							
	1	Reserved						
	Format: MBZ							
	0	ACE Enable						
	Format: Enable							
1	31:24	Y3						
		Default Value: 76						
		Format: U8						
	The value of the y_pixel for point 3 in PWL.							
	23:16	Y2						
		Default Value: 56						
		Format: U8						
	The value of the y_pixel for point 2 in PWL.							
	15:8	Y1						
		Default Value: 36						
		Format: U8						
	The value of the y_pixel for point 1 in PWL.							
7:0	Ymin							
	Default Value: 16							
	Format: U8							
The value of the y_pixel for point 0 in PWL.								
2	31:24	Y7						
		Default Value: 156						
		Format: U8						
	The value of the y_pixel for point 7 in PWL.							
	23:16	Y6						
		Default Value: 136						
Format: U8								
The value of the y_pixel for point 6 in PWL.								

VEBOX_ACE_LACE_STATE						
	15:8	<p>Y5</p> <table border="1"> <tr> <td>Default Value:</td> <td>116</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the y_pixel for point 5 in PWL.</p>	Default Value:	116	Format:	U8
	Default Value:	116				
Format:	U8					
	7:0	<p>Y4</p> <table border="1"> <tr> <td>Default Value:</td> <td>96</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the y_pixel for point 4 in PWL.</p>	Default Value:	96	Format:	U8
Default Value:	96					
Format:	U8					
3	31:24	<p>Ymax</p> <table border="1"> <tr> <td>Default Value:</td> <td>235</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the y_pixel for point 11 in PWL.</p>	Default Value:	235	Format:	U8
	Default Value:	235				
	Format:	U8				
	23:16	<p>Y10</p> <table border="1"> <tr> <td>Default Value:</td> <td>216</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the y_pixel for point 10 in PWL.</p>	Default Value:	216	Format:	U8
	Default Value:	216				
Format:	U8					
15:8	<p>Y9</p> <table border="1"> <tr> <td>Default Value:</td> <td>196</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the y_pixel for point 9 in PWL.</p>	Default Value:	196	Format:	U8	
Default Value:	196					
Format:	U8					
7:0	<p>Y8</p> <table border="1"> <tr> <td>Default Value:</td> <td>176</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the y_pixel for point 8 in PWL.</p>	Default Value:	176	Format:	U8	
Default Value:	176					
Format:	U8					
4	31:24	<p>B4</p> <table border="1"> <tr> <td>Default Value:</td> <td>96</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 4 in PWL.</p>	Default Value:	96	Format:	U8
	Default Value:	96				
Format:	U8					
23:16	<p>B3</p> <table border="1"> <tr> <td>Default Value:</td> <td>76</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 3 in PWL.</p>	Default Value:	76	Format:	U8	
Default Value:	76					
Format:	U8					

VEBOX_ACE_LACE_STATE						
	15:8	<p>B2</p> <table border="1"> <tr> <td>Default Value:</td> <td>56</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 2 in PWL.</p>	Default Value:	56	Format:	U8
	Default Value:	56				
Format:	U8					
	7:0	<p>B1</p> <table border="1"> <tr> <td>Default Value:</td> <td>36</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 1 in PWL.</p>	Default Value:	36	Format:	U8
Default Value:	36					
Format:	U8					
5	31:24	<p>B8</p> <table border="1"> <tr> <td>Default Value:</td> <td>176</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 8 in PWL.</p>	Default Value:	176	Format:	U8
	Default Value:	176				
	Format:	U8				
	23:16	<p>B7</p> <table border="1"> <tr> <td>Default Value:</td> <td>156</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 7 in PWL.</p>	Default Value:	156	Format:	U8
Default Value:	156					
Format:	U8					
15:8	<p>B6</p> <table border="1"> <tr> <td>Default Value:</td> <td>136</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 6 in PWL.</p>	Default Value:	136	Format:	U8	
Default Value:	136					
Format:	U8					
7:0	<p>B5</p> <table border="1"> <tr> <td>Default Value:</td> <td>116</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 5 in PWL.</p>	Default Value:	116	Format:	U8	
Default Value:	116					
Format:	U8					
6	31:16	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
15:8	<p>B10</p> <table border="1"> <tr> <td>Default Value:</td> <td>216</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>The value of the bias for point 10 in PWL.</p>	Default Value:	216	Format:	U8	
Default Value:	216					
Format:	U8					

VEBOX_ACE_LACE_STATE				
	7:0	B9 Default Value: 196 Format: U8 The value of the bias for point 9 in PWL.		
		<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
7	31:27	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	26:16	S1 Default Value: 1024 Format: U1.10 The value of the slope for point 1 in PWL The default is 1024/1024		
	15:11	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	10:0	S0 Default Value: 1024 Format: U1.10 The value of the slope for point 0 in PWL The default is 1024/1024		
	31:27	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
	Format:	MBZ		
	26:16	S3 Default Value: 1024 Format: U1.10 The value of the slope for point 3 in PWL The default is 1024/1024		
	15:11	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
10:0	S2 Default Value: 1024 Format: U1.10 The value of the slope for point 2 in PWL The default is 1024/1024			

VEBOX_ACE_LACE_STATE				
9	31:27	Reserved Format: MBZ		
	26:16	S5 Default Value: 1024 Format: U1.10 The value of the slope for point 5 in PWL The default is 1024/1024		
		15:11	Reserved Format: MBZ	
		10:0	S4 Default Value: 1024 Format: U1.10 The value of the slope for point 4 in PWL The default is 1024/1024	
			10	31:27
	26:16			S7 Default Value: 1024 Format: U1.10 The value of the slope for point 7 in PWL The default is 1024/1024
		15:11		Reserved Format: MBZ
		10:0		S6 Default Value: 1024 Format: U1.10 The default is 1024/1024
11	31:27		Reserved Format: MBZ	

VEBOX_ACE_LACE_STATE				
	26:16	S9	Default Value:	1024
			Format:	U1.10
			The value of the slope for point 9 in PWL	
			The default is 1024/1024	
	15:11	Reserved	Format:	MBZ
	10:0	S8	Default Value:	1024
			Format:	U1.10
			The value of the slope for point 8 in PWL	
			The default is 1024/1024	
	12	31:16	Max_ACE_luma	Format:
			The maximum luma for which ACE correction will be used.	
15:11		Reserved	Format:	MBZ
10:0		S10	Default Value:	1024
			Format:	U1.10
			The value of the slope for point 10 in PWL.	

VEBOX_ALPHA_AOI_STATE

VEBOX_ALPHA_AOI_STATE									
Source:	VideoEnhancementCS								
Size (in bits):	96								
Default Value:	0x00000000, 0x00030000, 0x00030000								
This state structure contains the IECP State Table Contents for Fixed Alpha State and Area of Interest State.									
DWord	Bit	Description							
0	31:18	Reserved Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">MBZ</td></tr></table>		MBZ					
		MBZ							
	17	Full Image Histogram Default Value: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">0</td></tr></table> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">Enable</td></tr></table> Used to ignore the area of interest for a histogram across the full image. This applies to all statistics that are affected by AOI (Area of Interest).		0		Enable			
		0							
	Enable								
16	Alpha from State Select Format: <table border="1" style="width: 100%;"><tr><td style="width: 40%;"></td><td style="width: 60%; text-align: center;">U1 Enumerated type</td></tr></table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%; text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>alpha is taken from message</td> </tr> <tr> <td style="text-align: center;">1</td> <td>alpha is taken from state</td> </tr> </tbody> </table> <div style="text-align: center; background-color: #e6f2ff; padding: 5px;">Programming Notes</div> If the input format does not have alpha available and the output format provides alpha, this bit should be set to 1. This should be 0 when Alpha Plane Enable is 1.		U1 Enumerated type	Value	Name	0	alpha is taken from message	1	alpha is taken from state
	U1 Enumerated type								
Value	Name								
0	alpha is taken from message								
1	alpha is taken from state								
15:0	Color Pipe Alpha Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U16</td></tr></table> <div style="text-align: center; background-color: #e6f2ff; padding: 5px;">Programming Notes</div> The 8 MSB of this field will be used for output formats that have 8-bits of alpha.		U16						
	U16								
1	31:30	Reserved Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">MBZ</td></tr></table>		MBZ					
			MBZ						
(This row is merged with the previous one in the original image)									

VEBOX_ALPHA_AOI_STATE	
	29:16 AOI Max X
	Default Value: 3
	Format: U14
	Area of Interest Minimum X - The ACE histogram and Skin Tone Detection statistic gathering will occur within the MinX/MinY to MaxX/MaxY area (inclusive). AOI must intersect the frame such that at least 1 pixel is in the AOI.
	The Area of Interest applies to the RGB Histogram and the White/Gray point sums as well.
	Programming Notes
	This value must be a multiple of 4 minus 1.
	15:14 Reserved
	Format: MBZ
	13:0 AOI Min X
Default Value: 0	
Format: U14	
Programming Notes	
This value must be a multiple of 4.	
2	31:30 Reserved
	Format: MBZ
	29:16 AOI Max Y
	Default Value: 3
	Format: U14
	Programming Notes
	This value must be a multiple of 4 minus 1.
	15:14 Reserved
	Format: MBZ
	13:0 AOI Min Y
Default Value: 0	
Format: U14	
Programming Notes	
This value must be a multiple of 4.	

VEBOX_CAPTURE_PIPE_STATE

VEBOX_CAPTURE_PIPE_STATE								
Source:	VideoEnhancementCS							
Size (in bits):	224							
Default Value:	0x0511FF23, 0xAA64AFAA, 0xE6FD4000, 0x00000000, 0x00000000, 0x00000000, 0x00000000							
This command contains variables for controlling Demosaic and the White Balance Statistics.								
DWord	Bit	Description						
0	31:30	Reserved Format: MBZ						
	29:24	Good Pixel Threshold Format: U6 The difference threshold between adjacent pixels for a pixel to be considered "good". <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>5h</td> <td>[Default]</td> </tr> </tbody> </table>	Value	Name	5h	[Default]		
	Value	Name						
	5h	[Default]						
	23	Reserved Format: MBZ						
	22:20	Shift Min Cost Default Value: 1h Format: U3 The amount to shift the H2/V2 versions of min_cost.						
	19:16	Green Imbalance Threshold Default Value: 1h Format: U4						
	15:8	Average Color Threshold Format: U8 The threshold between two colors in a pixel for the Avg interpolation to be considered. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>FFh</td> <td>[Default]</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>Must be set to 255.</td> </tr> </tbody> </table>	Value	Name	FFh	[Default]	Programming Notes	Must be set to 255.
	Value	Name						
	FFh	[Default]						
Programming Notes								
Must be set to 255.								
7:6	Reserved Format: U2							

VEBOX_CAPTURE_PIPE_STATE						
	5:0	<p>Good Pixel Neighbor Threshold</p> <table border="1"> <tr> <td>Default Value:</td> <td>23h</td> </tr> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Number of comparisons with neighbor pixels which pass before a pixel is considered good.</p>	Default Value:	23h	Format:	U6
	Default Value:	23h				
Format:	U6					
1	31:28	<p>Scale For Min Cost</p> <table border="1"> <tr> <td>Default Value:</td> <td>Ah</td> </tr> </table> <p>The amount to scale the min_cost difference during the confidence check.</p>	Default Value:	Ah		
	Default Value:	Ah				
	27:24	<p>Good Intesity Threshold</p> <table border="1"> <tr> <td>Default Value:</td> <td>Ah</td> </tr> <tr> <td>Format:</td> <td>U4</td> </tr> </table>	Default Value:	Ah	Format:	U4
	Default Value:	Ah				
	Format:	U4				
	23:16	<p>Bad Color Threshold 1</p> <table border="1"> <tr> <td>Default Value:</td> <td>64h</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Color value threshold used during the bad pixel check.</p>	Default Value:	64h	Format:	U8
	Default Value:	64h				
Format:	U8					
15:8	<p>Bad Color Threshold 2</p> <table border="1"> <tr> <td>Default Value:</td> <td>AFh</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Color value threshold used during the bad pixel check.</p>	Default Value:	AFh	Format:	U8	
Default Value:	AFh					
Format:	U8					
7:4	<p>Number Big Pixel Threshold</p> <table border="1"> <tr> <td>Default Value:</td> <td>Ah</td> </tr> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Number of comparisons with neighbor pixels which pass before a pixel is considered good.</p>	Default Value:	Ah	Format:	U4	
Default Value:	Ah					
Format:	U4					
3:0	<p>Bad Color Threshold 3</p> <table border="1"> <tr> <td>Default Value:</td> <td>Ah</td> </tr> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Color value threshold used during the bad pixel check.</p>	Default Value:	Ah	Format:	U4	
Default Value:	Ah					
Format:	U4					
2	31:24	<p>Y Bright Value</p> <table border="1"> <tr> <td>Default Value:</td> <td>E6h</td> </tr> </table> <p>The whitepoint threshold percentile in the Y histogram. Any pixel with Y value above this could be a whitepoint. This is the larger of the calculated Ybright value and the Ythreshold value, which is the minimum Y required to be considered a white point.</p>	Default Value:	E6h		
	Default Value:	E6h				
	Programming Notes					
"00000000" is appended to the LSBs before comparing with Y.						

VEBOX_CAPTURE_PIPE_STATE											
23:16	<p>Y Outlier Value</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">FDh</td> </tr> </table> <p>The outlier threshold percentile in the Y histogram. Any pixel with Y value above this either clipped or an outlier in the image. These points will not be included in the white patch calculation.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>"00000000" is appended to the LSBs before comparing with Y.</p>	Default Value:	FDh	Programming Notes							
	Default Value:	FDh									
	Programming Notes										
	15:8	<p>UV Threshold Value</p> <p>The value denotes the maximum threshold of the ratio between U+V to Y can have to be considered a gray point.</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>[255,0]</td> <td></td> <td>Encode a value from 255/256 to 0/256</td> </tr> <tr> <td>64</td> <td>[Default]</td> <td>0.25 * 255 = 64</td> </tr> </tbody> </table>	Value	Name	Description	[255,0]		Encode a value from 255/256 to 0/256	64	[Default]	0.25 * 255 = 64
		Value	Name	Description							
		[255,0]		Encode a value from 255/256 to 0/256							
		64	[Default]	0.25 * 255 = 64							
	7	Black Point Offset Red MSB									
	6	Black Point Offset Green Top MSB									
	5	Black Point Offset Blue MSB									
	4	Black Point Offset Green Bottom MSB									
	3	<p>RGB Histogram Enable</p> <p>Enables the collection of RGB Histograms for Auto-white balance correction and other uses.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>This bit can be set without White Balance enable being set.</p>	Programming Notes								
Programming Notes											
2	<p>Vignette Correction Format</p> <p>Defines what shift should be assumed for the Vignette Correction input values:</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>U8.8</td> </tr> <tr> <td>1</td> <td>U4.12</td> </tr> </tbody> </table>	Value	Name	0	U8.8	1	U4.12				
	Value	Name									
	0	U8.8									
1	U4.12										
1	<p>Black Point Correction Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table>	Format:	Enable								
	Format:	Enable									
0	<p>White Balance Correction Enable</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;">Programming Notes</td> </tr> </table> <p>RGB Histogram enable must be set if this bit is set.</p>	Format:	Enable	Programming Notes							
	Format:	Enable									
	Programming Notes										
3	31:16	<p>Black Point Offset Red</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">0</td> </tr> </table> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>Value subtracted from Red pixels of Bayer pattern - combined with MSB to form a 2's complement signed number.</p>	Default Value:	0	Format:	U16					
		Default Value:	0								
		Format:	U16								

VEBOX_CAPTURE_PIPE_STATE					
	15:0	Black Point Offset Green Top			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U16</td> </tr> </table> <p>Value subtracted from the top Green pixels of Bayer pattern (X=1, Y=0 for Bayer Pattern #1) - combined with MSB to form a 2's complement signed number.</p>		Default Value:	0	Format:
Default Value:	0				
Format:	U16				
4	31:16	Black Point Offset Blue			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U16</td> </tr> </table> <p>Value subtracted from Blue pixels of Bayer pattern - Combine with MSB to form a 2's complement signed number.</p>		Default Value:	0	Format:
Default Value:	0				
Format:	U16				
	15:0	Black Point Offset Green Bottom			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U16</td> </tr> </table> <p>Value subtracted from the bottom Green pixels of Bayer pattern (X=0, Y=1 for Bayer Pattern #1) - combined with MSB to form a 2's complement signed number.</p>		Default Value:	0	Format:
Default Value:	0				
Format:	U16				
5	31:16	White Balance Red Correction			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">U4.12</td> </tr> </table> <p>The correction factor multiplied by the Red pixels of the Bayer pattern.</p>		Format:	U4.12	
Format:	U4.12				
	15:0	White Balance Green Top Correction			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">U4.12</td> </tr> </table> <p>The correction factor multiplied by the top Green pixels of the Bayer pattern(X=1, Y=0 for Bayer Pattern #1).</p>		Format:	U4.12	
Format:	U4.12				
6	31:16	White Balance Blue Correction			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">U4.12</td> </tr> </table> <p>The correction factor multiplied by the Blue pixels of the Bayer pattern.</p>		Format:	U4.12	
Format:	U4.12				
	15:0	White Balance Green Bottom Correction			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">U4.12</td> </tr> </table> <p>The correction factor multiplied by the bottom Green pixels of the Bayer pattern (X=0, Y=1 for Bayer Pattern #1)</p>		Format:	U4.12	
Format:	U4.12				

VEBOX_CCM_STATE

VEBOX_CCM_STATE			
Source:	VideoEnhancementCS		
Size (in bits):	288		
Default Value:	0x00000475, 0x00000AE8, 0x00000047, 0x00000022, 0x0001FFCC, 0x00000D23, 0x000000A8, 0x0001FFF4, 0x00000D6A		
This state structure contains the IECP State Table Contents for the Color Correction Matrix State.			
DWord	Bit	Description	
0	31	Color Correction Matrix Enable Format: Enable This bit enables the Color Correction Matrix.	
		Programming Notes	
		Single Pipe IECP Enable must also be set if this bit is enabled.	
0	30:17	Reserved Format: MBZ	
		16:0	C1 Default Value: 000475h = 1141/4096 Format: S4.12 Coefficient of 3x3 Transform matrix
			Reserved Format: MBZ
1	16:0	C0 Default Value: 000AE8h = 2792/4096 Format: S4.12 Coefficient of 3x3 Transform matrix	
		31:17	Reserved Format: MBZ
			Reserved Format: MBZ
2	16:0	C3 Default Value: 000047h = 71/4096 Format: S4.12 Coefficient of 3x3 Transform matrix	
		31:17	Reserved Format: MBZ
			Reserved Format: MBZ
3	31:17	Reserved Format: MBZ	

VEBOX_CCM_STATE						
	16:0	C2 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>000022h = 34/4096</td> </tr> <tr> <td>Format:</td> <td>S4.12</td> </tr> </table> Coefficient of 3x3 Transform matrix	Default Value:	000022h = 34/4096	Format:	S4.12
Default Value:	000022h = 34/4096					
Format:	S4.12					
4	31:17	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
16:0	C5 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>1FFFCCh = -52/4096</td> </tr> <tr> <td>Format:</td> <td>S4.12</td> </tr> </table> Coefficient of 3x3 Transform matrix	Default Value:	1FFFCCh = -52/4096	Format:	S4.12	
Default Value:	1FFFCCh = -52/4096					
Format:	S4.12					
5	31:17	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
16:0	C4 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>000D23h = 3363/4096</td> </tr> <tr> <td>Format:</td> <td>S4.12</td> </tr> </table> Coefficient of 3x3 Transform matrix	Default Value:	000D23h = 3363/4096	Format:	S4.12	
Default Value:	000D23h = 3363/4096					
Format:	S4.12					
6	31:17	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
16:0	C7 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0000A8h = 168/4096</td> </tr> <tr> <td>Format:</td> <td>S4.12</td> </tr> </table> Coefficient of 3x3 Transform matrix	Default Value:	0000A8h = 168/4096	Format:	S4.12	
Default Value:	0000A8h = 168/4096					
Format:	S4.12					
7	31:17	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
16:0	C6 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>1FFFF4h = -12/4096</td> </tr> <tr> <td>Format:</td> <td>S4.12</td> </tr> </table> Coefficient of 3x3 Transform matrix	Default Value:	1FFFF4h = -12/4096	Format:	S4.12	
Default Value:	1FFFF4h = -12/4096					
Format:	S4.12					
8	31:17	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					

VEBOX_CCM_STATE						
	16:0	C8 <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td>000D6Ah = 3434/4096</td> </tr> <tr> <td>Format:</td> <td>S4.12</td> </tr> </table> Coefficient of 3x3 Transform matrix	Default Value:	000D6Ah = 3434/4096	Format:	S4.12
Default Value:	000D6Ah = 3434/4096					
Format:	S4.12					

VEBOX_Ch_Dir_Filter_Coefficient

VEBOX_Ch_Dir_Filter_Coefficient		
Source:	BSpec	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:56	Filter Coefficient[7] Format: <input type="text"/> S1.6 2's Complement Range: [-2, +2)
	55:48	Filter Coefficient[6] Format: <input type="text"/> S1.6 2's Complement Range: [-2, +2)
	47:40	Filter Coefficient[5] Format: <input type="text"/> S1.6 2's Complement Range: [-2, +2)
	39:32	Filter Coefficient[4] Format: <input type="text"/> S1.6 2's Complement Range: [-2, +2)
	31:24	Filter Coefficient[3] Format: <input type="text"/> S1.6 2's Complement Range: [-2, +2)
	23:16	Filter Coefficient[2] Format: <input type="text"/> S1.6 2's Complement Range: [-2, +2)
	15:8	Filter Coefficient[1] Format: <input type="text"/> S1.6 2's Complement Range: [-2, +2)
	7:0	Filter Coefficient[0] Format: <input type="text"/> S1.6 2's Complement Range: [-2, +2)

VEBOX_CSC_STATE

VEBOX_CSC_STATE																
Source:	VideoEnhancementCS															
Size (in bits):	384															
Default Value:	0x00010000, 0x00000000, 0x00000000, 0x00000000, 0x00010000, 0x00000000, 0x00000000, 0x00000000, 0x00010000, 0x00000000, 0x00000000, 0x00000000															
This state structure contains the IECP State Table Contents for CSC state.																
DWord	Bit	Description														
0	31	Transform Enable <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table>	Format:	Enable												
	Format:	Enable														
	30	YUV_Channel_Swap <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit should only be used with R8G8B8A8 output formats. When this bit is set, the YUV channels are swapped into the output RGB channels as shown in the following table to support B8G8R8A8 output on surface format R8G8B8A8:</p> <table border="1" style="width: 100%;"> <tr> <td></td> <td>YUV_Channel_Swap</td> </tr> <tr> <td></td> <td>0 --> 1</td> </tr> <tr> <td>Y</td> <td>R --> B</td> </tr> <tr> <td>U</td> <td>G --> G</td> </tr> <tr> <td>V</td> <td>B --> R</td> </tr> </table>	Default Value:	0	Format:	Enable		YUV_Channel_Swap		0 --> 1	Y	R --> B	U	G --> G	V	B --> R
		Default Value:	0													
		Format:	Enable													
		YUV_Channel_Swap														
	0 --> 1															
Y	R --> B															
U	G --> G															
V	B --> R															
Programming Notes																
This bit is to provide support for B8G8R8A8 output even though it is not supported in the surface format.																
29:19	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ													
Format:	MBZ															
18:0	C0 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>10000h or 1.0</td> </tr> <tr> <td>Format:</td> <td>S2.16 2's complement</td> </tr> </table> <p>Transform coefficient.</p>	Default Value:	10000h or 1.0	Format:	S2.16 2's complement											
Default Value:	10000h or 1.0															
Format:	S2.16 2's complement															
1	31:19	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ												
	Format:	MBZ														
18:0	C1 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S2.16 2's complement</td> </tr> </table> <p>Transform coefficient.</p>	Default Value:	0	Format:	S2.16 2's complement											
Default Value:	0															
Format:	S2.16 2's complement															

VEBOX_CSC_STATE		
2	31:19	Reserved Format: _____ MBZ
	18:0	C2 Default Value: _____ 0 Format: _____ S2.16 2's complement Transform coefficient.
3	31:19	Reserved Format: _____ MBZ
	18:0	C3 Default Value: _____ 0 Format: _____ S2.16 2's complement Transform coefficient.
4	31:19	Reserved Format: _____ MBZ
	18:0	C4 Default Value: _____ 10000h or 1.0 Format: _____ S2.16 2's complement Transform coefficient.
5	31:19	Reserved Format: _____ MBZ
	18:0	C5 Default Value: _____ 0 Format: _____ S2.16 2's complement Transform coefficient.
6	31:19	Reserved Format: _____ MBZ
	18:0	C6 Default Value: _____ 0 Format: _____ S2.16 2's complement Transform coefficient.
7	31:19	Reserved Format: _____ MBZ

VEBOX_CSC_STATE						
	18:0	<p>C7</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S2.16 2's complement</td> </tr> </table> <p>Transform coefficient.</p>	Default Value:	0	Format:	S2.16 2's complement
Default Value:	0					
Format:	S2.16 2's complement					
8	31:19	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
18:0	<p>C8</p> <table border="1"> <tr> <td>Default Value:</td> <td>10000h or 1.0</td> </tr> <tr> <td>Format:</td> <td>S2.16 2's complement</td> </tr> </table> <p>Transform coefficient. The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	10000h or 1.0	Format:	S2.16 2's complement	
Default Value:	10000h or 1.0					
Format:	S2.16 2's complement					
9	31:16	<p>Offset Out 1</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15 2's Complement</td> </tr> </table> <p>Offset in for Y/R. The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	0	Format:	S15 2's Complement
	Default Value:	0				
Format:	S15 2's Complement					
15:0	<p>Offset in 1</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15 2's Complement</td> </tr> </table> <p>Offset in for Y/R. The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	0	Format:	S15 2's Complement	
Default Value:	0					
Format:	S15 2's Complement					
10	31:16	<p>Offset Out 2</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15 2's Complement</td> </tr> </table> <p>Offset out for U/G. The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	0	Format:	S15 2's Complement
	Default Value:	0				
Format:	S15 2's Complement					
15:0	<p>Offset in 2</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15 2's Complement</td> </tr> </table> <p>Offset out for U/G. The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	0	Format:	S15 2's Complement	
Default Value:	0					
Format:	S15 2's Complement					
11	31:16	<p>Offset Out 3</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15 2's Complement</td> </tr> </table> <p>Offset out for V/B. The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	0	Format:	S15 2's Complement
Default Value:	0					
Format:	S15 2's Complement					

VEBOX_CSC_STATE					
15:0	<p>Offset in 3</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Default Value:</td> <td style="width: 50%; text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td>S15 2's Complement</td> </tr> </table> <p>Offset out for V/B. The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	0	Format:	S15 2's Complement
Default Value:	0				
Format:	S15 2's Complement				

VEBOX_DNDI_STATE

VEBOX_DNDI_STATE												
Source:	VideoEnhancementCS											
Size (in bits):	576											
Default Value:	0x00000800, 0x00000000, 0x00008000, 0x00010000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000254, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x005064A5											
<p>This state table is used by the <i>Denoise and Deinterlacer functions</i>.</p> <ul style="list-style-type: none"> • DW0 to 2 are for Temporal Denoise • DW3 is for global noise estimate and hot pixel detection • DW4 is for Chroma Denoise • DW5 to 11 are for 5x5 spatial denoise • DW12 to 17 are for Deinterlacer 												
DWord	Bit	Description										
0	31:20	Denoise STAD Threshold <table border="1"> <tr> <td>Format:</td> <td>U12</td> </tr> </table> Threshold for denoise sum of temporal absolute differences.	Format:	U12								
	Format:	U12										
	19:12	Denoise Maximum History <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Maximum allowed value for denoise history. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[128,240]</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	[128,240]					
	Format:	U8										
	Value	Name										
[128,240]												
11:8	Denoise History increase <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> Amount that denoise_history is increased by. MAX:15 <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>8h</td> <td>[Default]</td> <td></td> </tr> <tr> <td>15</td> <td></td> <td>Maximum Allowed</td> </tr> </tbody> </table>	Format:	U4	Value	Name	Description	8h	[Default]		15		Maximum Allowed
Format:	U4											
Value	Name	Description										
8h	[Default]											
15		Maximum Allowed										
7:5	Reserved <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ									
Format:	MBZ											
4:0	Denoise Moving Pixel Threshold <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16]</td> <td></td> </tr> </tbody> </table>	Format:	U5	Value	Name	[0,16]						
Format:	U5											
Value	Name											
[0,16]												

		VEBOX_DNDI_STATE		
1	31:20	Denoise ASD Threshold		
		Format:	U12	
		Threshold for denoise absolute sum of differences.		
		Value	Name	
		[0,1023]		
19:10	Temporal Difference Threshold			
		Format:	U10	
		Programming Notes		
		0 < (Temporal Difference Threshold - Low Temporal Difference Threshold) < = 256 except when both thresholds are set to 0.		
9:0	Low Temporal Difference Threshold			
		Format:	U10	
		Programming Notes		
		0 < (Temporal Difference Threshold - Low Temporal Difference Threshold) < = 256 except when both thresholds are set to 0.		
2	31:29	Reserved		
		Format:	MBZ	
	28	Progressive DN		
		Format:	Enable	
		Indicates that the denoise algorithm should assume progressive input when filtering neighboring pixels. This bit must be set if the input to Denoise is RGB.		
		Value	Name	Description
		0		DN assumes interlaced video and filters alternate lines together
		1		DN assumes progressive video and filters neighboring lines together
		Programming Notes		
		DI Enable must be disabled when this field is enabled.		
27:16	Denoise Threshold for Sum of Complexity Measure			
	Format:	U12		
15:10	Initial Denoise History			
	Default Value:	32		
	Format:	U6		
	Initial value for Denoise history for both Luma and Chroma			
9:0	Reserved			
	Format:	MBZ		

VEBOX_DNDI_STATE												
3	31:28	<p>Hot Pixel Count</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> </table> <p>Number of neighboring pixels different more than Hot Pixel Threshold before a pixel is considered hot.</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>[0,8]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Programming Notes</p> <p>0 will cause all pixels to be considered hot and will perform a median filter on the entire image.</p>	Format:	U4	Value	Name	[0,8]					
	Format:	U4										
	Value	Name										
	[0,8]											
27:20	<p>Hot Pixel Threshold</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>Threshold for a difference from the value of a neighboring pixel. Is shifted up to 16-bits before compare.</p>	Format:	U8									
Format:	U8											
19:12	<p>Block Noise Estimate Edge Threshold</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>Threshold for detecting an edge in block noise estimate.</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>16</td> <td>[Default]</td> <td></td> </tr> <tr> <td>255</td> <td></td> <td>Maximum Value</td> </tr> </tbody> </table>	Format:	U8	Value	Name	Description	16	[Default]		255		Maximum Value
Format:	U8											
Value	Name	Description										
16	[Default]											
255		Maximum Value										
11:0	<p>Block Noise Estimate Noise Threshold</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U12</td> </tr> </table> <p>Threshold for noise maximum/minimum.</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, 511]</td> <td></td> </tr> </tbody> </table>	Format:	U12	Value	Name	[0, 511]						
Format:	U12											
Value	Name											
[0, 511]												
4	31:24	<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										
	23:16	<p>Chroma Denoise STAD Threshold</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>Threshold for denoise sum of temporal absolute differences.</p>	Format:	U8								
	Format:	U8										
15:13	<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											
12	<p>Chroma Denoise Enable</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>1</td> <td></td> <td>The U and V chroma channels will be denoise filtered.</td> </tr> <tr> <td>0</td> <td></td> <td>The U and V channels will be passed to the next stage after DN unchanged.</td> </tr> </tbody> </table>	Value	Name	Description	1		The U and V chroma channels will be denoise filtered.	0		The U and V channels will be passed to the next stage after DN unchanged.		
Value	Name	Description										
1		The U and V chroma channels will be denoise filtered.										
0		The U and V channels will be passed to the next stage after DN unchanged.										

VEBOX_DNDI_STATE		
	11:6	Chroma Temporal Difference Threshold Format: U6 Programming Notes $0 < (\text{Chroma Temporal Difference Threshold} - \text{Chroma Low Temporal Difference Threshold}) \leq 16$ (Larger than 0 and less than or equal to 16)
	5:0	Chroma Low Temporal Difference Threshold Format: U6 Programming Notes $0 < (\text{Chroma Temporal Difference Threshold} - \text{Chroma Low Temporal Difference Threshold}) \leq 16$
5	31:30	Reserved Format: MBZ
	29:25	Dn_Wr5 $th4 \leq$ (Weight to be applied to pixels with differences in pixel value)
	24:20	Dn_Wr4 $th3 \leq$ (Weight to be applied to pixels with differences in pixel value) < $th4$
	19:15	Dn_Wr3 $th2 \leq$ (Weight to be applied to pixels with differences in pixel value) < $th3$
	14:10	Dn_Wr2 $th1 \leq$ (Weight to be applied to pixels with differences in pixel value) < $th2$
	9:5	Dn_Wr1 $th0 \leq$ (Weight to be applied to pixels with differences in pixel value) < $th1$
	4:0	Dn_Wr0 (Weight to be applied to pixels with differences in pixel value) < $th0$
6	31:29	Reserved Format: MBZ
	28:16	Dn_thmax Maximum threshold value
	15:13	Reserved Format: MBZ
	12:0	Dn_thmin Minimum threshold value
7	31:29	Reserved Format: MBZ
	28:16	Dn_prt5
	15:13	Reserved Format: MBZ

VEBOX_DNDI_STATE				
	12:0	Dn_dyn_thmin Minimum Dynamic threshold value.		
8	31:29	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
	28:16	Dn_prt4 Multiplied by thrscale and then used as the threshold for comparing the pixel differences.		
	15:13	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>MBZ</td></tr></table>		MBZ
	MBZ			
12:0	Dn_prt3			
9	31:29	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
	28:16	Dn_prt2		
	15:13	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>MBZ</td></tr></table>		MBZ
	MBZ			
12:0	Dn_prt1			
10	31:29	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
	28:16	Dn_prt0		
	15	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
	14:10	Dn_wd22 Weight to be applied to the 4 pixels that are at X±2 and Y±2		
9:5	Dn_wd21 Weight to be applied to the 4 pixels that are at X±1 and Y±2			
4:0	Dn_wd20 Weight to be applied to the 2 pixels that are at X and Y±2			
11	31:30	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;"> </td><td>MBZ</td></tr></table>		MBZ
		MBZ		
	29:25	Dn_wd12 Weight to be applied to the 4 pixels that are at X±2 and Y±1		
	24:20	Dn_wd11 Weight to be applied to the 4 pixels that are at X±1 and Y±1		
	19:15	Dn_wd10 Weight to be applied to the 2 pixels that are at X and Y±1		
	14:10	Dn_wd02 Weight to be applied to the 2 pixels that are at X±2 and Y		
9:5	Dn_wd01 Weight to be applied to the 2 pixels that are at X±1 and Y			

VEBOX_DNDI_STATE											
	4:0	Dn_wd00 Weight to be applied to the 1 pixels that are at X and Y									
12	31:13	Reserved Format: MBZ									
		STMM C2 Format: U3 Bias for divisor in STMM equation.									
		<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>Representing values [1,8]</td> </tr> </tbody> </table>	Value	Name	Description	[0,7]		Representing values [1,8]			
	Value	Name	Description								
	[0,7]		Representing values [1,8]								
	9:6	Content Adaptive Threshold Slope Format: U4 Determines the slope of the Content Adaptive Threshold.									
			<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>9</td> <td>[Default]</td> <td>CAT_slope value = 10</td> </tr> </tbody> </table>	Value	Name	Description	9	[Default]	CAT_slope value = 10		
		Value	Name	Description							
		9	[Default]	CAT_slope value = 10							
		Programming Notes									
+1 added internally to get CAT_slope.											
5:2	SAD Tight Threshold Default Value: 5 Format: U4										
	Smooth MV Threshold Format: U2										
	STMM Blending Constant Select Format: U1										
13	31	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Use the blending constant for small values of STMM for stmm_md_th</td> </tr> <tr> <td>1</td> <td></td> <td>Use the blending constant for large values of STMM for stmm_md_th</td> </tr> </tbody> </table>	Value	Name	Description	0		Use the blending constant for small values of STMM for stmm_md_th	1		Use the blending constant for large values of STMM for stmm_md_th
		Value	Name	Description							
		0		Use the blending constant for small values of STMM for stmm_md_th							
	1		Use the blending constant for large values of STMM for stmm_md_th								
	30:24	Blending constant across time for large values of STMM Format: U7									
		Blending constant across time for small values of STMM Format: U8									
	15:14	Reserved Format: MBZ									
		13:8	Multiplier for VECM Format: U6 Determines the strength of the vertical edge complexity measure.								

VEBOX_DNDI_STATE														
14	7:0	<p>Maximum STMM</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>Largest allowed STMM in blending equations.</p>	Format:	U8										
	Format:	U8												
	31:24	<p>Minimum STMM</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>Smallest allowed STMM in blending equations</p>	Format:	U8										
	Format:	U8												
	23:22	<p>STMM Shift Down</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Amount to shift STMM down (quantize to fewer bits)</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>Shift by 4</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Shift by 5</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Shift by 6</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Reserved</td> </tr> </tbody> </table>	Format:	U2	Value	Name	0	Shift by 4	1	Shift by 5	2	Shift by 6	3	Reserved
	Format:	U2												
	Value	Name												
	0	Shift by 4												
	1	Shift by 5												
	2	Shift by 6												
	3	Reserved												
	21:20	<p>STMM Shift Up</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Amount to shift STMM up (set range).</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>Shift by 6</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Shift by 7</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Shift by 8</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Reserved</td> </tr> </tbody> </table>	Format:	U2	Value	Name	0	Shift by 6	1	Shift by 7	2	Shift by 8	3	Reserved
	Format:	U2												
	Value	Name												
0	Shift by 6													
1	Shift by 7													
2	Shift by 8													
3	Reserved													
19:16	<p>STMM Output Shift</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</td> </tr> </table> <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;">[0,16]</td> <td></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>The value of this field must satisfy the following equation: $stmm_max - stmm_min = 2 ^{stmm_output_shift}$</td> </tr> </tbody> </table>	Format:	U4	Value	Name	[0,16]		Programming Notes	The value of this field must satisfy the following equation: $stmm_max - stmm_min = 2 ^{stmm_output_shift}$					
Format:	U4													
Value	Name													
[0,16]														
Programming Notes														
The value of this field must satisfy the following equation: $stmm_max - stmm_min = 2 ^{stmm_output_shift}$														
15:8	<p>SDI Threshold</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>Threshold for angle detection in SDI algorithm.</p>	Format:	U8											
Format:	U8													

VEBOX_DNDI_STATE											
	7:0	SDI Delta Format: U8 Delta value for angle detection in SDI algorithm.									
		SDI Fallback Mode 1 T1 Constant Format: U8									
15	31:24	SDI Fallback Mode 1 T1 Constant Format: U8									
	23:16	SDI Fallback Mode 1 T2 Constant Format: U8									
	15:8	SDI Fallback Mode 2 Constant (Angle2x1) Format: U8									
	7:0	FMD Temporal Difference Threshold Format: U8									
16	31:24	FMD #1 Vertical Difference Threshold Format: U8									
	23:16	FMD #2 Vertical Difference Threshold Format: U8									
	15:14	CAT Threshold Default Value: 0 Format: U2									
	13:8	FMD Tear Threshold Format: U6									
	7	MCDI Enable Use Motion Compensated Deinterlace algorithm. <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> This bit is Ignored if DI Enable is off.									
	6:4	Reserved Format: MBZ									
	3	DN/DI Top First Format: Enable Indicates the top field is first in sequence, otherwise bottom is first. <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>Bottom field occurs first in sequence</td> </tr> <tr> <td>1</td> <td></td> <td>Top field occurs first in sequence</td> </tr> </tbody> </table>	Value	Name	Description	0		Bottom field occurs first in sequence	1		Top field occurs first in sequence
	Value	Name	Description								
	0		Bottom field occurs first in sequence								
	1		Top field occurs first in sequence								
2:0	Reserved Format: MBZ										
17	31:26	Reserved Format: MBZ									

VEBOX_DNDI_STATE			
25:23	Reserved		
	Format:	MBZ	
22:19	Neighbor Pixel Threshold		
	Default Value:	10	
	Format:	U4	
18	Reserved		
	Format:	MBZ	
17:16	Progressive Cadence Reconstruction For 2nd Field Of Previous Frame		
	Format:	U2	
	Value	Name	Description
	0	Deinterlace	
	1	Put together with previous field in sequence	1 st field of previous frame
	2	Put together with next field in sequence	1 st field of current frame
15:10	MC Pixel Consistency Threshold		
	Default Value:	25	
	Format:	U6	
9:8	Progressive Cadence Reconstruction for 1st Field of Current Frame		
	Format:	U2	
	Value	Name	Description
	0	Deinterlace	
	1	Put together with previous field in sequence	2 nd field of previous frame
	2	Put together with next field in sequence	2 nd field of current frame
7:4	SAD THB		
	Default Value:	10	
	Format:	U4	
3:0	SAD THA		
	Default Value:	5	
	Format:	U4	



VEBOX_Filter_Coefficient

VEBOX_Filter_Coefficient				
Source:	BSpec			
Size (in bits):	8			
Default Value:	0x00000000			
DWord	Bit	Description		
0	7:0	2's Complement Filter Coefficient <table border="1" data-bbox="500 611 1468 655"> <tr> <td>Format:</td> <td>S1.6 2's Complement</td> </tr> </table> Range: [-2, +2)	Format:	S1.6 2's Complement
Format:	S1.6 2's Complement			

VEBOX_FRONT_END_CSC_STATE

VEBOX_FRONT_END_CSC_STATE		
Source:	VideoEnhancementCS	
Size (in bits):	384	
Default Value:	0x00010000, 0x00000000, 0x00000000, 0x00000000, 0x00010000, 0x00000000, 0x00000000, 0x00000000, 0x00010000, 0x00000000, 0x00000000, 0x00000000	
This state structure contains the IECP State Table Contents for Front-end CSC state.		
DWord	Bit	Description
0	31	Front End CSC Transform Enable Format: Enable
		Programming Notes Single Pipe IECP Enable must also be set if this is enabled.
		Reserved Format: MBZ
1	18:0	FECS C0: Transform coefficient Default Value: 10000h or 1.0 Format: S2.16
		Reserved Format: MBZ
		FECS C1: Transform coefficient Default Value: 0 or 0.0 Format: S2.16
2	18:0	FECS C2: Transform coefficient Default Value: 0 or 0.0 Format: S2.16
		Reserved Format: MBZ
		FECS C3: Transform coefficient Default Value: 0 or 0.0 Format: S2.16
3	18:0	Reserved Format: MBZ
		Reserved Format: MBZ
		Reserved Format: MBZ
4	31:19	Reserved Format: MBZ
		Reserved Format: MBZ
		Reserved Format: MBZ

VEBOX_FRONT_END_CSC_STATE		
	18:0	FECSC C4: Transform coefficient
		Default Value: 10000h or 1.0
		Format: S2.16
5	31:19	Reserved
		Format: MBZ
		18:0 FECSC C5: Transform coefficient
		Default Value: 0 or 0.0
		Format: S2.16
6	31:19	Reserved
		Format: MBZ
		18:0 FECSC C6: Transform coefficient
		Default Value: 0 or 0.0
		Format: S2.16
7	31:19	Reserved
		Format: MBZ
		18:0 FECSC C7: Transform coefficient
		Default Value: 0 or 0.0
		Format: S2.16
8	31:19	Reserved
		Format: MBZ
		18:0 FECSC C8: Transform coefficient
		Default Value: 10000h or 1.0
		Format: S2.16
9	31:16	FEC SC Offset out 1: Offset out for Y/R
		Default Value: 0
		Format: S15
	The offset value is multiplied by 2 before being added to the output.	
	15:0 FEC SC Offset in 1: Offset in for Y/R	
		Default Value: 0
		Format: S15
The offset value is multiplied by 2 before being added to the output.		
10	31:16	FEC SC Offset out 2: Offset out for U/G
		Default Value: 0
		Format: S15
The offset value is multiplied by 2 before being added to the output.		

VEBOX_FRONT_END_CSC_STATE						
	15:0	<p>FEC SC Offset in 2: Offset out for U/G</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15</td> </tr> </table> <p>The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	0	Format:	S15
Default Value:	0					
Format:	S15					
11	31:16	<p>FEC SC Offset out 3: Offset out for V/B</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15</td> </tr> </table> <p>The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	0	Format:	S15
	Default Value:	0				
Format:	S15					
	15:0	<p>FEC SC Offset in 3: Offset out for V/B</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15</td> </tr> </table> <p>The offset value is multiplied by 2 before being added to the output.</p>	Default Value:	0	Format:	S15
Default Value:	0					
Format:	S15					

VEBOX_GAMUT_STATE

VEBOX_GAMUT_STATE								
Source:		VideoEnhancementCS						
Size (in bits):		1216						
Default Value:		0x01B40000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x09050201, 0x412A1A10, 0x00BB8860, 0x3526170D, 0x8B725B47, 0x00DFC1A5, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x654F371E, 0x00000000, 0x00EDDBC8, 0x21140A03, 0x755C4331, 0x00D7B493, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x0CD2911F, 0xB0000334, 0x00000000						
DWord	Bit	Description						
0	31:25	Reserved Format: MBZ						
	24:16	A(r) Default Value: 436 Format: U9 Gain_factor_R (default: 436, preferred range: 256-511)						
	15	Global Mode Enable The gain factor derived from state CM(w) <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>Advance Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Basic Mode</td> </tr> </tbody> </table>	Value	Name	0	Advance Mode	1	Basic Mode
	Value	Name						
	0	Advance Mode						
	1	Basic Mode						
14:10	Reserved Format: MBZ							
9:0	CM(w) Format: U10 WeightingFactorForGain_factor (only enabled when the GlobalModeEnable is on)							
1	31:26	Reserved Format: MBZ						
	25:16	CM(s) Format: U2.8 AccurateColorComponentScaling (default: 640/256, preferred range: [512-1023]/256) The default is 640/256						

VEBOX_GAMUT_STATE			
	15	Reserved Format: MBZ	
	14:8	A(g) Format: U7 Gain_factor_G (default: 26/256, preferred range: [26-127]/256) The default is 26/256	
		7	Reserved Format: MBZ
		6:0	A(b) Format: U7 Gain_factor_B (default: 26/256, preferred range: [26-127]/256) The default is 26/256
	2	31:26	Reserved Format: MBZ
		25:16	R(s) Format: U2.8 RedScaling (default: 768/256, preferred range: [512-1023]/256) The default is 768/256
15:8			CM(i) Format: U0.8 AccurateColorComponentOffset (default: 192/256, preferred range: [0-192]/256) The default is 192/256
7:0		R(i) Format: U0.8 RedOffset (default: 128/256, preferred range: [0-128]/256) The default is 128/256	
3		31	Reserved Format: MBZ
		30:16	C1 Format: S2.12 Coefficient of 3x3 Transform matrix The default is 1141/4096

VEBOX_GAMUT_STATE			
	15	Reserved Format: _____ MBZ	
	14:0	C0 Format: _____ S2.12 Coefficient of 3x3 Transform matrix The default is 2792/4096	
		31	Reserved Format: _____ MBZ
		30:16	C3 Format: _____ S2.12 Coefficient of 3x3 Transform matrix The default is 71/4096
4	15	Reserved Format: _____ MBZ	
	14:0	C2 Format: _____ S2.12 Coefficient of 3x3 Transform matrix The default is 34/4096	
		31	Reserved Format: _____ MBZ
		30:16	C5 Format: _____ S2.12 Coefficient of 3x3 Transform matrix The default is -52/4096
5	15	Reserved Format: _____ MBZ	
	14:0	C4 Format: _____ S2.12 Coefficient of 3x3 Transform matrix The default is 3663/4096	
		31	Reserved Format: _____ MBZ
		30:16	C5 Format: _____ S2.12 Coefficient of 3x3 Transform matrix The default is -52/4096
6	31	Reserved Format: _____ MBZ	

VEBOX_GAMUT_STATE								
	30:16	<p>C7</p> <table border="1"> <tr> <td>Format:</td> <td>S2.12</td> </tr> <tr> <td colspan="2">Coefficient of 3x3 Transform matrix</td> </tr> <tr> <td colspan="2">The default is 168/4096</td> </tr> </table>	Format:	S2.12	Coefficient of 3x3 Transform matrix		The default is 168/4096	
	Format:	S2.12						
	Coefficient of 3x3 Transform matrix							
The default is 168/4096								
15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ					
Format:	MBZ							
14:0	<p>C6</p> <table border="1"> <tr> <td>Format:</td> <td>S2.12</td> </tr> <tr> <td colspan="2">Coefficient of 3x3 Transform matrix</td> </tr> <tr> <td colspan="2">The default is -12/4096</td> </tr> </table>	Format:	S2.12	Coefficient of 3x3 Transform matrix		The default is -12/4096		
Format:	S2.12							
Coefficient of 3x3 Transform matrix								
The default is -12/4096								
7	31:15	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ						
14:0	<p>C8</p> <table border="1"> <tr> <td>Format:</td> <td>S2.12</td> </tr> <tr> <td colspan="2">Coefficient of 3x3 Transform matrix</td> </tr> <tr> <td colspan="2">The default is 3434/4096</td> </tr> </table>	Format:	S2.12	Coefficient of 3x3 Transform matrix		The default is 3434/4096		
Format:	S2.12							
Coefficient of 3x3 Transform matrix								
The default is 3434/4096								
8	31:24	<p>PWL_Gamma_Point 4</p> <table border="1"> <tr> <td>Default Value:</td> <td>9</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> <tr> <td colspan="2">Point 4 for PWL for gamma correction</td> </tr> </table>	Default Value:	9	Format:	U8	Point 4 for PWL for gamma correction	
		Default Value:	9					
		Format:	U8					
	Point 4 for PWL for gamma correction							
	23:16	<p>PWL_Gamma_Point 3</p> <table border="1"> <tr> <td>Default Value:</td> <td>5</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> <tr> <td colspan="2">Point 3 for PWL for gamma correction</td> </tr> </table>	Default Value:	5	Format:	U8	Point 3 for PWL for gamma correction	
		Default Value:	5					
		Format:	U8					
	Point 3 for PWL for gamma correction							
	15:8	<p>PWL_Gamma_Point 2</p> <table border="1"> <tr> <td>Default Value:</td> <td>2</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> <tr> <td colspan="2">Point 2 for PWL for gamma correction</td> </tr> </table>	Default Value:	2	Format:	U8	Point 2 for PWL for gamma correction	
		Default Value:	2					
		Format:	U8					
	Point 2 for PWL for gamma correction							
7:0	<p>PWL_Gamma_Point 1</p> <table border="1"> <tr> <td>Default Value:</td> <td>1</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> <tr> <td colspan="2">Point 1 for PWL for gamma correction</td> </tr> </table>	Default Value:	1	Format:	U8	Point 1 for PWL for gamma correction		
	Default Value:	1						
	Format:	U8						
Point 1 for PWL for gamma correction								

VEBOX_GAMUT_STATE		
9	31:24	PWL_Gamma_Point 8
		Default Value: 65 Point 8 for PWL for gamma correction
	23:16	PWL_Gamma_Point 7
		Default Value: 42 Point 7 for PWL for gamma correction
15:8	PWL_Gamma_Point 6	
	Default Value: 26 Point 6 for PWL for gamma correction	
7:0	PWL_Gamma_Point 5	
	Default Value: 16 Point 5 for PWL for gamma correction	
10	31:24	Reserved
		Format: MBZ
	23:16	PWL_Gamma_Point 11
		Default Value: 187 Format: U8 Point 11 for PWL for gamma correction
15:8	PWL_Gamma_Point 10	
	Default Value: 136 Format: U8 Point 10 for PWL for gamma correction	
7:0	PWL_Gamma_Point 9	
	Default Value: 96 Format: U8 Point 9 for PWL for gamma correction	
11	31:24	PWL_Gamma_Bias_4
		Default Value: 53
		Format: U8 Bias 4 for PWL for gamma correction

VEBOX_GAMUT_STATE		
	23:16	PWL_Gamma_Bias_3 Default Value: 38 Format: U8 Bias 3 for PWL for gamma correction
		PWL_Gamma_Bias_2 Default Value: 23 Format: U8 Bias 2 for PWL for gamma correction
		PWL_Gamma_Bias_1 Default Value: 13 Format: U8 Bias 1 for PWL for gamma correction
12	31:24	PWL_Gamma_Bias_8 Default Value: 139 Format: U8 Bias 8 for PWL for gamma correction
		PWL_Gamma_Bias_7 Default Value: 114 Format: U8 Bias 7 for PWL for gamma correction
		PWL_Gamma_Bias_6 Default Value: 91 Format: U8 Bias 6 for PWL for gamma correction
		PWL_Gamma_Bias_5 Default Value: 71 Format: U8 Bias 5 for PWL for gamma correction
13	31:24	Reserved Format: MBZ

VEBOX_GAMUT_STATE		
	23:16	PWL_Gamma_Bias_11
		Default Value: 223
		Format: U8
		Bias 11 for PWL for gamma correction
	15:8	PWL_Gamma_Bias_10
		Default Value: 193
		Format: U8
		Bias 10 for PWL for gamma correction
	7:0	PWL_Gamma_Bias_9
Default Value: 165		
Format: U8		
Bias 9 for PWL for gamma correction		
14	31:28	Reserved
		Format: MBZ
	27:16	PWL_Gamma_Slope_1
		Format: U4.8
		Slope 1 for PWL for gamma correction
		The default is 2560/256
	15:12	Reserved
		Format: MBZ
11:0	PWL_Gamma_Slope_0	
	Format: U4.8	
	Slope 0 for PWL for gamma correction	
	The default is 3328/256	
15	31:28	Reserved
		Format: MBZ
	27:16	PWL_Gamma_Slope_3
		Format: U4.8
		Slope 3 for PWL for gamma correction
		The default is 960/256
	15:12	Reserved
Format: MBZ		

VEBOX_GAMUT_STATE		
	11:0	PWL_Gamma_Slope_2
		Format: U4.8
		Slope 2 for PWL for gamma correction The default is 1280/256
16	31:28	Reserved
		Format: MBZ
	27:16	PWL_Gamma_Slope_5
		Format: U4.8
		Slope 5 for PWL for gamma correction The default is 512/256
	15:12	Reserved
		Format: MBZ
	11:0	PWL_Gamma_Slope_4
		Format: U4.8
		Slope 4 for PWL for gamma correction The default is 658/256
17	31:28	Reserved
		Format: MBZ
	27:16	PWL_Gamma_Slope_7
		Format: U4.8
		Slope 7 for PWL for gamma correction The default is 278/256
	15:12	Reserved
		Format: MBZ
	11:0	PWL_Gamma_Slope_6
		Format: U4.8
		Slope 6 for PWL for gamma correction The default is 368/256
18	31:28	Reserved
		Format: MBZ

VEBOX_GAMUT_STATE		
	27:16	PWL_Gamma_Slope_9
		Format: U4.8
		Slope 9 for PWL for gamma correction The default is 179/256
	15:12	Reserved
		Format: MBZ
		11:0
Format: U4.8		
Slope 8 for PWL for gamma correction The default is 215/256		
19	31:28	Reserved
		Format: MBZ
	27:16	PWL_Gamma_Slope_11
		Format: U4.8
		Slope 11 for PWL for gamma correction The default is 124/256
	15:12	Reserved
		Format: MBZ
	11:0	PWL_Gamma_Slope_10
Format: U4.8		
Slope 10 for PWL for gamma correction The default is 151/256		
20	31:24	PWL_INV_GAMMA_Point 4
		Default Value: 101
		Format: U8 Point 4 for PWL for inverse gamma correction
	23:16	PWL_INV_GAMMA_Point 3
		Default Value: 79
		Format: U8 Point 3 for PWL for inverse gamma correction

VEBOX_GAMUT_STATE								
	15:8	PWL_INV_GAMMA_Point 2 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">55</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Point 2 for PWL for inverse gamma correction	Default Value:	55	Format:	U8		
	Default Value:	55						
Format:	U8							
7:0	PWL_INV_GAMMA_Point 1 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">30</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Point 1 for PWL for inverse gamma correction	Default Value:	30	Format:	U8			
Default Value:	30							
Format:	U8							
21	31:24	PWL_INV_GAMMA_Point 8 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Point 8 for PWL for inverse gamma correction <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;">181</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	181	
	Format:	U8						
	Value	Name						
	181							
23:16	PWL_INV_GAMMA_Point 7 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Point 7 for PWL for inverse gamma correction <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;">162</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	162		
Format:	U8							
Value	Name							
162								
15:8	PWL_INV_GAMMA_Point 6 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Point 6 for PWL for inverse gamma correction <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;">141</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	141		
Format:	U8							
Value	Name							
141								
7:0	PWL_INV_GAMMA_Point 5 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Point 5 for PWL for inverse gamma correction <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;">122</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	122		
Format:	U8							
Value	Name							
122								
22	31:24	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ						
23:16	PWL_INV_GAMMA_Point 11 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">237</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> Point 11 for PWL for inverse gamma correction	Default Value:	237	Format:	U8			
Default Value:	237							
Format:	U8							

VEBOX_GAMUT_STATE		
	15:8	PWL_INV_GAMMA_Point 10 Default Value: 219 Format: U8 Point 10 for PWL for inverse gamma correction
		PWL_INV_GAMMA_Point 9 Default Value: 200 Format: U8 Point 9 for PWL for inverse gamma correction
23	31:24	PWL_INV_GAMMA_Bias 4 Default Value: 33 Format: U8 Bias 4 for PWL for inverse gamma correction
		PWL_INV_GAMMA_Bias 3 Default Value: 20 Format: U8 Bias 3 for PWL for inverse gamma correction
	15:8	PWL_INV_GAMMA_Bias 2 Default Value: 10 Format: U8 Bias 2 for PWL for inverse gamma correction
		PWL_INV_GAMMA_Bias 1 Default Value: 3 Format: U8 Bias 1 for PWL for inverse gamma correction
24	31:24	PWL_INV_GAMMA_Bias 8 Default Value: 117 Format: U8 Bias 8 for PWL for inverse gamma correction
		PWL_INV_GAMMA_Bias 7 Default Value: 92 Format: U8 Bias 7 for PWL for inverse gamma correction

VEBOX_GAMUT_STATE						
	15:8	PWL_INV_GAMMA_Bias_6 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">67</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> Bias 6 for PWL for inverse gamma correction	Default Value:	67	Format:	U8
	Default Value:	67				
Format:	U8					
	7:0	PWL_INV_GAMMA_Bias_5 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">49</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> Bias 5 for PWL for inverse gamma correction	Default Value:	49	Format:	U8
Default Value:	49					
Format:	U8					
25	31:24	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	23:16	PWL_INV_GAMMA_Bias_11 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">215</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> Bias 11 for PWL for inverse gamma correction	Default Value:	215	Format:	U8
	Default Value:	215				
Format:	U8					
15:8	PWL_INV_GAMMA_Bias_10 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">180</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> Bias 10 for PWL for inverse gamma correction	Default Value:	180	Format:	U8	
Default Value:	180					
Format:	U8					
7:0	PWL_INV_GAMMA_Bias_9 <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">147</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> Bias 9 for PWL for inverse gamma correction	Default Value:	147	Format:	U8	
Default Value:	147					
Format:	U8					
26	31:28	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	27:16	PWL_INV_GAMMA_Slope_1 <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">U4.8</td> </tr> </table> Slope 1 for PWL for gamma correction The default is 72/256	Format:	U4.8		
Format:	U4.8					
15:12	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					

VEBOX_GAMUT_STATE		
	11:0	PWL_INV_GAMMA_Slope_0
		Format: U4.8
		Slope 0 for PWL for gamma correction The default is 26/256
27	31:28	Reserved
		Format: MBZ
	27:16	PWL_INV_GAMMA_Slope_3
		Format: U4.8
		Slope 3 for PWL for gamma correction The default is 151/256
	15:12	Reserved
		Format: MBZ
	11:0	PWL_INV_GAMMA_Slope_2
		Format: U4.8
		Slope 2 for PWL for gamma correction The default is 107/256
28	31:28	Reserved
		Format: MBZ
	27:16	PWL_INV_GAMMA_Slope_5
		Format: U4.8
		Slope 5 for PWL for gamma correction The default is 243/256
	15:12	Reserved
		Format: MBZ
	11:0	PWL_INV_GAMMA_Slope_4
		Format: U4.8
		Slope 4 for PWL for gamma correction The default is 195/256
29	31:28	Reserved
		Format: MBZ

VEBOX_GAMUT_STATE		
	27:16	PWL_INV_GAMMA_Slope_7 Format: U4.8 Slope 7 for PWL for gamma correction The default is 337/256
	15:12	Reserved Format: MBZ
	11:0	PWL_INV_GAMMA_Slope_6 Format: U4.8 Slope 6 for PWL for gamma correction The default is 305/256
30	31:28	Reserved Format: MBZ
	27:16	PWL_INV_GAMMA_Slope_9 Format: U4.8 Slope 9 for PWL for gamma correction The default is 445/256
	15:12	Reserved Format: MBZ
	11:0	PWL_INV_GAMMA_Slope_8 Format: U4.8 Slope 8 for PWL for gamma correction The default is 404/256
31	31:28	Reserved Format: MBZ
	27:16	PWL_INV_GAMMA_Slope_11 Format: U4.8 Slope 11 for PWL for gamma correction The default is 555/256
	15:12	Reserved Format: MBZ

VEBOX_GAMUT_STATE		
	11:0	PWL_INV_GAMMA_Slope_10 Format: U4.8 Slope 10 for PWL for gamma correction The default is 498/256
	32	31 Reserved Format: MBZ
	30:16	Offset_in_G Default Value: 0 Format: S14 The input offset for green component
	15	Reserved Format: MBZ
	14:0	Offset_in_R Default Value: 0 Format: S14 The input offset for red component
	33	31 Reserved Format: MBZ
	30:16	Offset_out_B Format: S2.12 The input offset for green component The default is -1246/4096
	15	Reserved Format: MBZ
	14:0	Offset_in_B Default Value: 0 Format: S14 The input offset for red component
	34	31 Reserved Format: MBZ

VEBOX_GAMUT_STATE								
	30:16	Offset_out_G <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S2.12</td> </tr> </table> <p>The input offset for green component The default is -983/4096</p>	Format:	S2.12				
	Format:	S2.12						
	15	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ							
14:0	Offset_out_R <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S2.12</td> </tr> </table> <p>The input offset for red component The default is -974/4096</p>	Format:	S2.12					
Format:	S2.12							
35	31	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
	Format:	MBZ						
	30	FullRangeMappingEnable <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>0</td> <td>Basic Mode [Default]</td> </tr> <tr> <td>1</td> <td>Advance Mode</td> </tr> </tbody> </table>	Value	Name	0	Basic Mode [Default]	1	Advance Mode
	Value	Name						
	0	Basic Mode [Default]						
1	Advance Mode							
29:20	d(in,default) <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>205</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> <p>InnerTriangleMappingLength</p>	Default Value:	205	Format:	U10			
Default Value:	205							
Format:	U10							
19:10	d(out, default) <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>164</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> <p>OuterTriangleMappingLength</p>	Default Value:	164	Format:	U10			
Default Value:	164							
Format:	U10							
9:0	d1(out) <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>287</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> <p>OuterTriangleMappingLengthBelow</p>	Default Value:	287	Format:	U10			
Default Value:	287							
Format:	U10							
36	31 xvYccDecEncEnable This bit is valid only when ColorGamutCompressionnEnable is on. <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>1</td> <td>Both xvYcc decode and xvYcc encode are enabled [Default]</td> </tr> <tr> <td>0</td> <td>To disable both xvYcc decode and xvYcc encode</td> </tr> </tbody> </table>	Value	Name	1	Both xvYcc decode and xvYcc encode are enabled [Default]	0	To disable both xvYcc decode and xvYcc encode	
Value	Name							
1	Both xvYcc decode and xvYcc encode are enabled [Default]							
0	To disable both xvYcc decode and xvYcc encode							

VEBOX_GAMUT_STATE																										
	30:28	CompressionLineShift <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>3</td> <td>[Default]</td> </tr> <tr> <td>[0,4]</td> <td></td> </tr> </tbody> </table>	Value	Name	3	[Default]	[0,4]																			
	Value	Name																								
	3	[Default]																								
	[0,4]																									
27:10	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																							
Format:	MBZ																									
9:0	d1(in) <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>820</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> InnerTriangleMappingLengthBelow	Default Value:	820	Format:	U10																					
Default Value:	820																									
Format:	U10																									
37	31:30 GCC BasicModeSelection <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>00b</td> <td>Default</td> <td></td> </tr> <tr> <td>01b</td> <td>Scaling Factor</td> <td>Used along with Dword66 Bits 28:11</td> </tr> <tr> <td>10b</td> <td>Single Axis Gamma Correction</td> <td>Used along with Dword67 Bit 29</td> </tr> <tr> <td>11b</td> <td>Scaling factor with fixed luma</td> <td>Used along with Dword37 Bits 28:11</td> </tr> </tbody> </table> 29 LumaChormaOnlyCorrection <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 75%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Luma Only Correction [Default]</td> </tr> <tr> <td>1</td> <td>Chorma Only Correction</td> </tr> </tbody> </table> 28:25 Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table> 24:11 BasicModeScalingFactor <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U2.12</td> </tr> </table> Used when FullRangeMappingEnable is in basic mode and base mode selection bit is set to scaling factor.	Value	Name	Description	00b	Default		01b	Scaling Factor	Used along with Dword66 Bits 28:11	10b	Single Axis Gamma Correction	Used along with Dword67 Bit 29	11b	Scaling factor with fixed luma	Used along with Dword37 Bits 28:11	Value	Name	0	Luma Only Correction [Default]	1	Chorma Only Correction	Format:	MBZ	Format:	U2.12
Value	Name	Description																								
00b	Default																									
01b	Scaling Factor	Used along with Dword66 Bits 28:11																								
10b	Single Axis Gamma Correction	Used along with Dword67 Bit 29																								
11b	Scaling factor with fixed luma	Used along with Dword37 Bits 28:11																								
Value	Name																									
0	Luma Only Correction [Default]																									
1	Chorma Only Correction																									
Format:	MBZ																									
Format:	U2.12																									
	10:1	Reserved <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ																						
Format:	MBZ																									
	0	Cpi Override <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 35%;">Value</th> <th style="width: 65%;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>[Default]</td> </tr> <tr> <td>1</td> <td>Override Cpi calculation</td> </tr> </tbody> </table>	Value	Name	0	[Default]	1	Override Cpi calculation																		
Value	Name																									
0	[Default]																									
1	Override Cpi calculation																									

VEBOX_IECP_STATE		
67..69	95:0	Alpha/AOI State
		Format: VEBOX_ALPHA_AOI_STATE For description of this state, refer to <i>Alpha State [Section]</i> .
70..78	287:0	CCM State
		Format: VEBOX_CCM_STATE For description of this state, refer to the CCM State section
79..90	383:0	Front-end CSC
		Format: VEBOX_FRONT_END_CSC_STATE For description of this state, refer to <i>Front-end CSC State Section</i> .

VEBOX_PROCAMP_STATE

VEBOX_PROCAMP_STATE		
Source:	VideoEnhancementCS	
Size (in bits):	64	
Default Value:	0x01000001, 0x01000000	
This state structure contains the IECP State Table Contents for ProcAmp state.		
DWord	Bit	Description
0	31:28	Reserved Format: MBZ
	27:17	Contrast Default Value: 80h = 1.0 in fixed point U4.7 Format: U4.7 Contrast magnitude.
	16:13	Reserved Format: MBZ
	12:1	Brightness Default Value: 0 or 0.0 Format: S7.4 2's complement Brightness magnitude.
	0	PROCAMP Enable Default Value: 1 Format: Enable
1	31:16	Cos_c_s Default Value: 256 Format: S7.8 2's complement UV multiplication cosine factor.
	15:0	Sin_c_s Default Value: 0 Format: S7.8 2's complement UV multiplication sine factor.

VEBOX_RGB_TO_GAMMA_CORRECTION

VEBOX_RGB_TO_GAMMA_CORRECTION			
Source:	VideoEnhancementCS		
Size (in bits):	64		
Default Value:	0x00000000, 0x00000000		
Color depth is 16 bits.			
DWord	Bit	Description	
0..1	63:48	B-ch Corrected Value	
		Default Value:	0h
		Format:	U16
	47:32	G-ch Corrected Value	
		Default Value:	0h
		Format:	U16
	31:16	R-ch Corrected Value	
		Default Value:	0h
		Format:	U16
	15:0	Pixel Value	
		Default Value:	0h
		Format:	U16
		Programming Notes	
		N indicates the index into the table. Pixel value 0 and Pixel Value 63 should be always programmed to 0 and 0xFFFF respectively. Rest of the Pixel Values which are used for comparing with the input pixel value should be multiple of 4.	

VEBOX_STD_STE_STATE

VEBOX_STD_STE_STATE			
Source:	VideoEnhancementCS		
Size (in bits):	928		
Default Value:	0x9A6E39F0, 0x400D3C65, 0x000C9180, 0xFE2F2E00, 0x0003FFFF, 0x00140000, 0xD82E0640, 0x8285ECEC, 0x07FB8282, 0x00000000, 0x02117000, 0xA38FEC96, 0x0100C8C8, 0x003A6871, 0x01478000, 0x0107C306, 0x1291F008, 0x00094855, 0x1C1BD100, 0x03802008, 0x0002A980, 0x00080180, 0x0007CFF5, 0x18D1F07C, 0x000800BD, 0x1C080100, 0x03800000, 0x0008012B, 0x0008012B		
This state structure contains the state used by the STD/STE function.			
DWord	Bit	Description	
0	31:24	V_Mid	
		Default Value:	154
		Format:	U8
		Rectangle middle-point V coordinate.	
	23:16	U_Mid	
		Default Value:	110
		Format:	U8
		Rectangle middle-point U coordinate.	
	15:10	Hue_Max	
		Default Value:	14
		Format:	U6
		Rectangle half width.	
	9:4	Sat_Max	
		Default Value:	31
		Format:	U6
		Rectangle half length.	
	3	Reserved	
		Format:	MBZ
	2	Output Control	
		Value	Name
0		Output Pixels	
1		Output STD Decisions	

VEBOX_STD_STE_STATE		
	1	STE Enable Format: _____ Enable
	0	STD Enable Format: _____ Enable <div style="text-align: center; background-color: #e6f2ff; padding: 2px;">Programming Notes</div> This needs to be enabled if 'STD Score Output' is enabled.
1	31	STD Score Output Format: _____ Enable
	30:28	Diamond Margin Default Value: _____ 4 Format: _____ U3
	27:21	Diamond_du Default Value: _____ 0 Format: _____ S6 2's complement Rhombus center shift in the sat-direction, relative to the rectangle center.
	20:18	HS_margin Default Value: _____ 3 Format: _____ U3 Defines rectangle margin.
	17:10	Cos(α) Default Value: _____ 79 Format: _____ S0.7 2's complement The default is 79/128
	9:8	Reserved Format: _____ MBZ
	7:0	Sin(α) Default Value: _____ 101 Format: _____ S0.7 2's complement The default is 101/128
2	31:21	Reserved Format: _____ MBZ

VEBOX_STD_STE_STATE						
	20:13	<p>Diamond_alpha</p> <table border="1"> <tr> <td>Default Value:</td> <td>100</td> </tr> <tr> <td>Format:</td> <td>U2.6</td> </tr> </table> <p>$1/\tan(\beta)$ The default is 100/64</p>	Default Value:	100	Format:	U2.6
	Default Value:	100				
	Format:	U2.6				
12:7	<p>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					
6:0	<p>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					
3	31:24	<p>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				
	23:16	<p>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					
15:8	<p>Y_point_1</p> <table border="1"> <tr> <td>Default Value:</td> <td>46</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>First point of the Y piecewise linear membership function.</p>	Default Value:	46	Format:	U8	
Default Value:	46					
Format:	U8					
7	<p>VY_STD_Enable</p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Enables STD in the VY subspace.</p>	Format:	Enable			
Format:	Enable					
6:0	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
4	31:18	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					

VEBOX_STD_STE_STATE		
	17:13	Y_Slope_2
		Default Value: 31
		Format: U2.3
		Slope between points Y3 and Y4.
		The default is 31/8
	12:8	Y_Slope_1
		Default Value: 31
		Format: U2.3
		Slope between points Y1 and Y2.
		The default is 31/8
7:0	Y_point_4	
	Default Value: 255	
	Format: U8	
Fourth point of the Y piecewise linear membership function.		
5	31:16	INV_Skin_types_margin
		Default Value: 20 Skin_Type_margin
	Format: U0.16	
	1/(2* Skin_types_margin)	
15:0	INV_Margin_VYL	
	Format: U0.16	
1 / Margin_VYL 1/ Margin_VYL = 3300/65536		
6	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.		

VEBOX_STD_STE_STATE						
	15:0	<p>INV_Margin_VYU</p> <table border="1"> <tr> <td>Default Value:</td> <td>1600</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>1 / Margin_VYU = 1600/65536</p>	Default Value:	1600	Format:	U0.16
Default Value:	1600					
Format:	U0.16					
7	31:24	<p>B1L</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 1 of the lower part of the detection PWLF.</p>	Default Value:	130	Format:	U8
	Default Value:	130				
	Format:	U8				
	23:16	<p>B0L</p> <table border="1"> <tr> <td>Default Value:</td> <td>133</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 0 of the lower part of the detection PWLF.</p>	Default Value:	133	Format:	U8
Default Value:	133					
Format:	U8					
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					
8	31:27	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	26:16	<p>S0L</p> <table border="1"> <tr> <td>Default Value:</td> <td>FFBh</td> </tr> <tr> <td>Format:</td> <td>S2.8 2's complement</td> </tr> </table> <p>Slope 0 of the lower part of the detection PWLF. The default is -5/256</p>	Default Value:	FFBh	Format:	S2.8 2's complement
Default Value:	FFBh					
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					

VEBOX_STD_STE_STATE						
	7:0	B2L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">130</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">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					
9	31:22	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	21:11	S2L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">S2.8 2's complement</td> </tr> </table> <p>The default is 0/256</p>	Default Value:	0	Format:	S2.8 2's complement
Default Value:	0					
Format:	S2.8 2's complement					
10:0	S1L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">S2.8 2's complement</td> </tr> </table> <p>Slope 1 of the lower part of the detection PWLF. The default is 0/256</p>	Default Value:	0	Format:	S2.8 2's complement	
Default Value:	0					
Format:	S2.8 2's complement					
10	31:27	Reserved <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	26:19	P1U <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">66</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> <p>Y Point 1 of the upper part of the detection PWLF.</p>	Default Value:	66	Format:	U8
	Default Value:	66				
Format:	U8					
18:11	P0U <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">46</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> <p>Y Point 0 of the upper part of the detection PWLF.</p>	Default Value:	46	Format:	U8	
Default Value:	46					
Format:	U8					
10:0	S3L <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">S2.8 2's complement</td> </tr> </table> <p>Slope 3 of the lower part of the detection PWLF. The default is 0/256</p>	Default Value:	0	Format:	S2.8 2's complement	
Default Value:	0					
Format:	S2.8 2's complement					

		VEBOX_STD_STE_STATE		
11	31:24	B1U		
		Default Value:	163	
		Format:	U8	
	V Bias 1 of the upper part of the detection PWLF.			
	23:16	B0U		
		Default Value:	143	
		Format:	U8	
	V Bias 0 of the upper part of the detection PWLF.			
	15:8	P3U		
		Default Value:	236	
		Format:	U8	
	Y Point 3 of the upper part of the detection PWLF.			
	7:0	P2U		
		Default Value:	150	
		Format:	U8	
	Y Point 2 of the upper part of the detection PWLF.			
12	31:27	Reserved		
		Format:	MBZ	
	26:16	S0U		
		Default Value:	256	
		Format:	S2.8 2's complement	
		Slope 0 of the upper part of the detection PWLF. The default is 256/256		
	15:8	B3U		
		Default Value:	200	
		Format:	U8	
		V Bias 3 of the upper part of the detection PWLF.		
7:0	B2U			
	Default Value:	200		
	Format:	U8		
	V Bias 2 of the upper part of the detection PWLF.			

VEBOX_STD_STE_STATE			
13	31:22	Reserved	
		Format: MBZ	
	21:11	S2U	
		Default Value: F4Dh	
		Format: S2.8 2's complement	
		Slope 2 of the upper part of the detection PWLF.	
		The default is -179/256	
	10:0	S1U	
		Default Value: 113	
		Format: S2.8	
Slope 1 of the upper part of the detection PWLF.			
The default is 113/256			
14	31:28	Reserved	
		Format: MBZ	
	27:20	Skin_types_margin	
		Default Value: 20	
		Format: U8	
	Skin types Y margin Restrict Skin_types_thresh >= Skin_types_margin > 0 Restrict (Skin_types_thresh + Skin_types_margin) <= 255		
	19:12	Skin_types_thresh	
		Default Value: 120	
		Format: U8	
	Skin types Y margin Restrict Skin_types_thresh >= Skin_types_margin > 0 Restrict (Skin_types_thresh + Skin_types_margin) <= 255		
	11	Skin_Types_Enable	
		Default Value: 0 Disable	
		Format: Enable	
	Treat differently bright and dark skin types		
	10:0	S3U	
Default Value: 0			
Format: S2.8 2's complement			
Slope 3 of the upper part of the detection PWLF.			
The default is 0/256			

VEBOX_STD_STE_STATE			
15	31	Reserved Format: _____ MBZ	
	30:21	SATB1 Default Value: _____ 8 Format: _____ S7.2 2's complement First bias for the saturation PWLF (bright skin). The default is 8/4	
		20:14	SATP3 Default Value: _____ 31 Format: _____ S6 2's complement Third point for the saturation PWLF (bright skin).
			13:7
		6:0	
16	31	Reserved Format: _____ MBZ	
	30:20	SATS0 Default Value: _____ 297 Format: _____ U3.8 Zeroth slope for the saturation PWLF (bright skin) The default is 297/256	
		19:10	SATB3 Default Value: _____ 124 Format: _____ S7.2 2's complement Third bias for the saturation PWLF (bright skin) The default is 124/4

		VEBOX_STD_STE_STATE	
	9:0	SATB2	
		Default Value:	8
		Format:	S7.2 2's complement
		Second bias for the saturation PWLF (bright skin) The default is 8/4	
17	31:22	Reserved	
		Format:	MBZ
	21:11	SATS2	
		Default Value:	297
		Format:	U3.8
		Second slope for the saturation PWLF (bright skin) The default is 297/256	
	10:0	SATS1	
		Default Value:	85
		Format:	U3.8
		First slope for the saturation PWLF (bright skin) The default is 85/256	
18	31:25	HUEP3	
		Default Value:	14
		Format:	S6 2's complement
	Third point for the hue PWLF (bright skin)		
	24:18	HUEP2	
		Default Value:	6
		Format:	S6 2's complement
	Second point for the hue PWLF (bright skin)		
	17:11	HUEP1	
Default Value:		7Ah -6	
Format:		S6 2's complement	
First point for the hue PWLF (bright skin)			

		VEBOX_STD_STE_STATE	
19	10:0	SATS3	
		Default Value:	256
		Format:	U3.8
		Third slope for the saturation PWLF (bright skin)	
		The default is 256/256	
	31:30	Reserved	
		Format:	MBZ
	29:20	HUEB3	
		Default Value:	56
		Format:	S7.2 2's complement
	Third bias for the hue PWLF (bright skin)		
	The default is 56/4		
19:10	HUEB2		
	Default Value:	8	
	Format:	S7.2 2's complement	
	Second bias for the hue PWLF (bright skin)		
	The default is 8/4		
9:0	HUEB1		
	Default Value:	8	
	Format:	S7.2 2's complement	
	First bias for the hue PWLF (bright skin)		
	The default is 8/4		
20	31:22	Reserved	
		Format:	MBZ
	21:11	HUES1	
		Default Value:	85
		Format:	U3.8
	First slope for the hue PWLF (bright skin)		
	The default is 85/256		

		VEBOX_STD_STE_STATE	
21	10:0	HUES0	
		Default Value:	384
		Format:	U3.8
		Zeroth slope for the hue PWLF (bright skin) The default is 384/256	
	31:22	Reserved	
		Format:	MBZ
	21:11	HUES3	
		Default Value:	256
		Format:	U3.8
		Third slope for the hue PWLF (bright skin) The default is 256/256	
10:0	HUES2		
	Default Value:	384	
	Format:	U3.8	
	Second slope for the hue PWLF (bright skin) The default is 384/256		
22	31	Reserved	
		Format:	MBZ
	30:21	SATB1_DARK	
		Default Value:	0
		Format:	S7.2 2's complement
		First bias for the saturation PWLF (dark skin) The default is 0/4	
	20:14	SATP3_DARK	
		Default Value:	31
		Format:	S6 2's complement
	Third point for the saturation PWLF (dark skin)		
13:7	SATP2_DARK		
	Default Value:	31	
	Format:	S6 2's complement	
Second point for the saturation PWLF (dark skin)			

VEBOX_STD_STE_STATE			
	6:0	SATP1_DARK Default Value: FF5h Format: S6 2's complement First point for the saturation PWLF (dark skin) Default Value: -11	
		Reserved Format: MBZ	
23	31	SATS0_DARK Default Value: 397 Format: U3.8 Zeroth slope for the saturation PWLF (dark skin) The default is 397/256	
		19:10	SATB3_DARK Default Value: 124 Format: S7.2 2's complement Third bias for the saturation PWLF (dark skin) The default is 124/4
			9:0
		31:22	
	21:11		SATS2_DARK Default Value: 256 Format: U3.8 Second slope for the saturation PWLF (dark skin) The default is 256/256

VEBOX_STD_STE_STATE					
	10:0	SATS1_DARK	Default Value:	189	
			Format:	U3.8	
			First slope for the saturation PWLF (dark skin)		
			The default is 189/256		
25	31:25	HUEP3_DARK	Default Value:	14	
			Format:	S6 2's complement	
			Third point for the hue PWLF (dark skin).		
	24:18	HUEP2_DARK	Default Value:	2	
			Format:	S6 2's complement	
			Second point for the hue PWLF (dark skin).		
	17:11	HUEP1_DARK	Default Value:	0	
			Format:	S6 2's complement	
			First point for the hue PWLF (dark skin).		
	10:0	SATS3_DARK	Default Value:	256	
			Format:	U3.8	
			Third slope for the saturation PWLF (dark skin)		
			The default is 256/256		
26	31:30	Reserved	Format:	MBZ	
	29:20	HUEB3_DARK	Default Value:	56	
			Format:	S7.2 2's complement	
			Third bias for the hue PWLF (dark skin).		
			The default is 56/4		

VEBOX_STD_STE_STATE		
	19:10	HUEB2_DARK
		Default Value: 0
		Format: S7.2 2's complement
		Second bias for the hue PWLF (dark skin).
		The default is 0/4
	9:0	HUEB1_DARK
		Default Value: 0
		Format: S7.2 2's complement
		First bias for the hue PWLF (dark skin).
		The default is 0/4
27	31:22	Reserved
		Format: MBZ
	21:11	HUES1_DARK
		Default Value: 256
		Format: U3.8
		First slope for the hue PWLF (dark skin).
		The default is 256/256
	10:0	HUES0_DARK
		Default Value: 299
Format: U3.8		
Zeroth slope for the hue PWLF (dark skin). The default is 299/256		
28	31:22	Reserved
		Format: MBZ
	21:11	HUES3_DARK
		Default Value: 256
		Format: U3.8
		Third slope for the hue PWLF (dark skin). The default is 256/256

VEBOX_STD_STE_STATE	
10:0	HUES2_DARK
	Default Value: 299
	Format: U3.8
	Second slope for the hue PWLF (dark skin).
	The default is 299/256

VEBOX_TCC_STATE

VEBOX_TCC_STATE			
Source:	VideoEnhancementCS		
Size (in bits):	352		
Default Value:	0xDCDCDC00, 0xDCDCDC00, 0x1E34CC91, 0x3E3CCE91, 0x02E80195, 0x0197046B, 0x01790174, 0x00096000, 0x00000000, 0x03030000, 0x009201C0		
This state structure contains the IECP State Table Contents for TCC state.			
DWord	Bit	Description	
0	31:24	SatFactor3	
		Default Value:	220
		Format:	U1.7
		The saturation factor for yellow. The default is 220/128	
	23:16	SatFactor2	
		Default Value:	220
		Format:	U1.7
		The saturation factor for red. The default is 220/128	
	15:8	SatFactor1	
		Default Value:	220
Format:		U1.7	
The saturation factor for magenta. The default is 220/128			
7	TCC Enable		
	Format:	Enable	
6:0	Reserved		
	Format:	MBZ	
1	31:24	SatFactor6	
		Default Value:	220
		Format:	U1.7
		The saturation factor for blue. The default is 220/128	

VEBOX_TCC_STATE		
	23:16	SatFactor5
		Default Value: 220
		Format: U1.7
		The saturation factor for cyan. The default is 220/128
	15:8	SatFactor4
		Default Value: 220
		Format: U1.7
		The saturation factor for green. The default is 220/128
	7:0	Reserved
		Format: MBZ
2	31:30	Reserved
		Format: MBZ
	29:20	BaseColor3
		Default Value: 483
		Format: U10
		Base Color 3 - this value must be greater than BaseColor2
	19:10	BaseColor2
		Default Value: 307
		Format: U10
Base Color 2 - this value must be greater than BaseColor1		
9:0	BaseColor1	
	Default Value: 145	
	Format: U10	
	Base Color 1	
3	31:30	Reserved
		Format: MBZ
	29:20	BaseColor6
		Default Value: 995 Format: U10 Base Color 6 - this value must be greater than BaseColor5

VEBOX_TCC_STATE						
	19:10	<p>BaseColor5</p> <table border="1"> <tr> <td>Default Value:</td> <td>819</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> <p>Base Color 5 - this value must be greater than BaseColor4</p>	Default Value:	819	Format:	U10
	Default Value:	819				
Format:	U10					
	9:0	<p>BaseColor4</p> <table border="1"> <tr> <td>Default Value:</td> <td>657</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> <p>Base Color 4 - this value must be greater than BaseColor3</p>	Default Value:	657	Format:	U10
Default Value:	657					
Format:	U10					
4	31:16	<p>ColorTransitSlope23</p> <table border="1"> <tr> <td>Default Value:</td> <td>744</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>The calculation result of $1 / (BC3 - BC2)$ [1/62]</p>	Default Value:	744	Format:	U0.16
	Default Value:	744				
Format:	U0.16					
	15:0	<p>ColorTransitSlope2</p> <table border="1"> <tr> <td>Default Value:</td> <td>405</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>The calculation result of $1 / (BC2 - BC1)$ [1/57]</p>	Default Value:	405	Format:	U0.16
Default Value:	405					
Format:	U0.16					
5	31:16	<p>ColorTransitSlope45</p> <table border="1"> <tr> <td>Default Value:</td> <td>407</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>The calculation result of $1 / (BC5 - BC4)$ [1/57]</p>	Default Value:	407	Format:	U0.16
	Default Value:	407				
Format:	U0.16					
	15:0	<p>ColorTransitSlope34</p> <table border="1"> <tr> <td>Default Value:</td> <td>1131</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>The calculation result of $1 / (BC4 - BC3)$ [1/61]</p>	Default Value:	1131	Format:	U0.16
Default Value:	1131					
Format:	U0.16					
6	31:16	<p>ColorTransitSlope61</p> <table border="1"> <tr> <td>Default Value:</td> <td>377</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>The calculation result of $1 / (BC1 - BC6)$ [1/62]</p>	Default Value:	377	Format:	U0.16
	Default Value:	377				
Format:	U0.16					
	15:0	<p>ColorTransitSlope56</p> <table border="1"> <tr> <td>Default Value:</td> <td>372</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p>The calculation result of $1 / (BC6 - BC5)$ [1/62]</p>	Default Value:	372	Format:	U0.16
Default Value:	372					
Format:	U0.16					

VEBOX_TCC_STATE						
7	31:22	<p>ColorBias3</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U2.8</td> </tr> </table> <p>Color bias for BaseColor3.</p>	Default Value:	0	Format:	U2.8
	Default Value:	0				
	Format:	U2.8				
	21:12	<p>ColorBias2</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="text-align: center;">150</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U2.8</td> </tr> </table> <p>Color bias for BaseColor2. The default is 150/256</p>	Default Value:	150	Format:	U2.8
Default Value:	150					
Format:	U2.8					
11:2	<p>ColorBias1</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U2.8</td> </tr> </table> <p>Color bias for BaseColor1.</p>	Default Value:	0	Format:	U2.8	
Default Value:	0					
Format:	U2.8					
1:0	<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					
8	31:22	<p>ColorBias6</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U2.8</td> </tr> </table> <p>Color bias for BaseColor6.</p>	Default Value:	0	Format:	U2.8
	Default Value:	0				
	Format:	U2.8				
	21:12	<p>ColorBias5</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U2.8</td> </tr> </table> <p>Color bias for BaseColor5.</p>	Default Value:	0	Format:	U2.8
Default Value:	0					
Format:	U2.8					
11:2	<p>ColorBias4</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U2.8</td> </tr> </table> <p>Color bias for BaseColor4.</p>	Default Value:	0	Format:	U2.8	
Default Value:	0					
Format:	U2.8					
1:0	<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	31	<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					

VEBOX_TCC_STATE						
	30:24	<p>UV Threshold</p> <table border="1"> <tr> <td>Default Value:</td> <td>3</td> </tr> <tr> <td>Format:</td> <td>U7</td> </tr> </table> <p>Low UV threshold.</p>	Default Value:	3	Format:	U7
	Default Value:	3				
	Format:	U7				
	23:19	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
	Format:	MBZ				
	18:16	<p>UV Threshold Bits</p> <table border="1"> <tr> <td>Default Value:</td> <td>3</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Low UV transition width bits.</p>	Default Value:	3	Format:	U3
	Default Value:	3				
Format:	U3					
15:13	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
12:8	<p>STE Threshold</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Skin tone pixels enhancement threshold.</p>	Default Value:	0	Format:	U5	
Default Value:	0					
Format:	U5					
7:3	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
2:0	<p>STE Slope Bits</p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Skin tone pixels enhancement slope bits.</p>	Default Value:	0	Format:	U3	
Default Value:	0					
Format:	U3					
10	31:16	<p>Inv_UVMaxColor</p> <table border="1"> <tr> <td>Default Value:</td> <td>146</td> </tr> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>1 / UVMaxColor. Used for the SFs2 calculation.</p>	Default Value:	146	Format:	U16
	Default Value:	146				
	Format:	U16				
15:9	<p>Reserved</p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ					
8:0	<p>UVMaxColor</p> <table border="1"> <tr> <td>Default Value:</td> <td>448</td> </tr> <tr> <td>Format:</td> <td>U9</td> </tr> </table> <p>The maximum absolute value of the legal UV pixels. Used for the SFs2 calculation.</p>	Default Value:	448	Format:	U9	
Default Value:	448					
Format:	U9					

VEBOX_VERTEX_TABLE								
0..511	31:28	Reserved						
		Format: MBZ						
	27:16	Vertex table entry 0 - Lv (12 bits)						
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%; text-align: center;">Value</th> <th style="width: 15%; text-align: center;">Name</th> <th style="width: 52%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">100h-ED6h</td> <td></td> <td>Range for Vertices BT601 and BT709</td> </tr> </tbody> </table>	Value	Name	Description	100h-ED6h		Range for Vertices BT601 and BT709
		Value	Name	Description				
	100h-ED6h		Range for Vertices BT601 and BT709					
	15:12	Reserved						
Format: MBZ								
11:0	Vertex table entry 0 - Cv (12 bits)							
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%; text-align: center;">Value</th> <th style="width: 15%; text-align: center;">Name</th> <th style="width: 52%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">400h-A00h</td> <td></td> <td>Range for Vertices BT601 and BT709</td> </tr> </tbody> </table>	Value	Name	Description	400h-A00h		Range for Vertices BT601 and BT709	
	Value	Name	Description					
400h-A00h		Range for Vertices BT601 and BT709						

VECS Hardware-Detected Error Bit Definitions

VECS Hardware-Detected Error Bit Definitions							
Source:	VideoEnhancementCS						
Size (in bits):	16						
Default Value:	0x00000000						
DWord	Bit	Description					
0	15:3	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ			
		MBZ					
	2	Command Privilege Violation Error This bit is set if a command classified as privileged is parsed in a non-privileged batch buffer. The command will be converted to a NOOP and parsing will continue.					
	1	Reserved Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		MBZ			
	MBZ						
0	Instruction Error This bit is set when the Renderer Instruction Parser detects an error while parsing an instruction. Instruction errors include: <ul style="list-style-type: none"> Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported). Defeated MI Instruction Opcodes: <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>1</td> <td></td> <td>Instruction Error detected</td> </tr> </tbody> </table>	Value	Name	Description	1		Instruction Error detected
Value	Name	Description					
1		Instruction Error detected					
Programming Notes							
This error indications cannot be cleared except by reset (i.e., it is a fatal error).							

VERTEX_BUFFER_STATE

VERTEX_BUFFER_STATE								
Source:	RenderCS							
Size (in bits):	128							
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000							
<p>This structure is used in 3DSTATE_VERTEX_BUFFERS to set the state associated with a VB. The VF function will use this state to determine how/where to extract vertex element data for all vertex elements associated with the VB.</p>								
DWord	Bit	Description						
0	31:26	Vertex Buffer Index						
		Format: U6 index						
		This field contains an index value which selects the VB state being defined.						
			<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,32]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,32]		
	Value	Name						
	[0,32]							
	25:23	Reserved						
		Format: MBZ						
	22:16	Memory Object Control State						
		Format: MEMORY_OBJECT_CONTROL_STATE						
15	Reserved							
	Format: MBZ							
14	Address Modify Enable							
	If set, the Buffer Starting Address field is used to update the state of this buffer. If clear, that field is ignored and the previously-programmed value is maintained.							
13	Null Vertex Buffer	Format: Enable						
		This field enabled causes any fetch for vertex data to return 0.						
	Programming Notes							
	VERTEX_BUFFER_STATE.Null Vertex Buffer must be set when the VERTEX_BUFFER_STATE.Buffer Size is 0x0.							
12	Reserved							
	Format: MBZ							
11:0	Buffer Pitch	Format: U12 Count of bytes						
		This field specifies the pitch in bytes of the structures accessed within the VB. This information is required in order to access elements in the VB via a structure index.						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,4095]</td> <td></td> <td>Bytes</td> </tr> </tbody> </table>	Value	Name	Description	[0,4095]		Bytes
	Value	Name	Description					
[0,4095]		Bytes						

VERTEX_BUFFER_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"> <ul style="list-style-type: none"> Different VERTEX_BUFFER_STATE structures can refer to the same memory region using different Buffer Pitch values. See note on 64-bit float alignment in Buffer Starting Address. </td> </tr> </tbody> </table>	Programming Notes		<ul style="list-style-type: none"> Different VERTEX_BUFFER_STATE structures can refer to the same memory region using different Buffer Pitch values. See note on 64-bit float alignment in Buffer Starting Address. 					
Programming Notes									
<ul style="list-style-type: none"> Different VERTEX_BUFFER_STATE structures can refer to the same memory region using different Buffer Pitch values. See note on 64-bit float alignment in Buffer Starting Address. 									
1..2	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center; vertical-align: top;">63:0</td> <td> <p>Buffer Starting Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>GraphicsAddress[63:0]Vertex_Buffer</td> </tr> </table> <p>This field contains the byte-aligned Graphics Address LSBs of the first element of interest within the VB. 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. If the Address ModifyEnable bit is clear, this field is ignored and the previous value of Buffer Starting Address for this buffer is maintained.</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"> <ul style="list-style-type: none"> 64-bit floating point values must be 64-bit aligned in memory, or UNPREDICTABLE data will be fetched. When accessing an element containing 64-bit floating point values, the Buffer Starting Address and Source Element Offset values must add to a 64-bit aligned address, and BufferPitch must be a multiple of 64-bits. VBs can only be allocated in linear (not tiled) graphics memory. As computed index values are, by definition, interpreted as unsigned values, there is no issue with accesses to locations before (lower address value) the start of the buffer. However, these wrapped indices are subject to Max Index checking (see below). </td> </tr> </tbody> </table> </td> </tr> </table>	63:0	<p>Buffer Starting Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>GraphicsAddress[63:0]Vertex_Buffer</td> </tr> </table> <p>This field contains the byte-aligned Graphics Address LSBs of the first element of interest within the VB. 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. If the Address ModifyEnable bit is clear, this field is ignored and the previous value of Buffer Starting Address for this buffer is maintained.</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"> <ul style="list-style-type: none"> 64-bit floating point values must be 64-bit aligned in memory, or UNPREDICTABLE data will be fetched. When accessing an element containing 64-bit floating point values, the Buffer Starting Address and Source Element Offset values must add to a 64-bit aligned address, and BufferPitch must be a multiple of 64-bits. VBs can only be allocated in linear (not tiled) graphics memory. As computed index values are, by definition, interpreted as unsigned values, there is no issue with accesses to locations before (lower address value) the start of the buffer. However, these wrapped indices are subject to Max Index checking (see below). </td> </tr> </tbody> </table>	Format:	GraphicsAddress[63:0]Vertex_Buffer	Programming Notes		<ul style="list-style-type: none"> 64-bit floating point values must be 64-bit aligned in memory, or UNPREDICTABLE data will be fetched. When accessing an element containing 64-bit floating point values, the Buffer Starting Address and Source Element Offset values must add to a 64-bit aligned address, and BufferPitch must be a multiple of 64-bits. VBs can only be allocated in linear (not tiled) graphics memory. As computed index values are, by definition, interpreted as unsigned values, there is no issue with accesses to locations before (lower address value) the start of the buffer. However, these wrapped indices are subject to Max Index checking (see below). 	
63:0	<p>Buffer Starting Address</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>GraphicsAddress[63:0]Vertex_Buffer</td> </tr> </table> <p>This field contains the byte-aligned Graphics Address LSBs of the first element of interest within the VB. 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. If the Address ModifyEnable bit is clear, this field is ignored and the previous value of Buffer Starting Address for this buffer is maintained.</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"> <ul style="list-style-type: none"> 64-bit floating point values must be 64-bit aligned in memory, or UNPREDICTABLE data will be fetched. When accessing an element containing 64-bit floating point values, the Buffer Starting Address and Source Element Offset values must add to a 64-bit aligned address, and BufferPitch must be a multiple of 64-bits. VBs can only be allocated in linear (not tiled) graphics memory. As computed index values are, by definition, interpreted as unsigned values, there is no issue with accesses to locations before (lower address value) the start of the buffer. However, these wrapped indices are subject to Max Index checking (see below). </td> </tr> </tbody> </table>	Format:	GraphicsAddress[63:0]Vertex_Buffer	Programming Notes		<ul style="list-style-type: none"> 64-bit floating point values must be 64-bit aligned in memory, or UNPREDICTABLE data will be fetched. When accessing an element containing 64-bit floating point values, the Buffer Starting Address and Source Element Offset values must add to a 64-bit aligned address, and BufferPitch must be a multiple of 64-bits. VBs can only be allocated in linear (not tiled) graphics memory. As computed index values are, by definition, interpreted as unsigned values, there is no issue with accesses to locations before (lower address value) the start of the buffer. However, these wrapped indices are subject to Max Index checking (see below). 			
Format:	GraphicsAddress[63:0]Vertex_Buffer								
Programming Notes									
<ul style="list-style-type: none"> 64-bit floating point values must be 64-bit aligned in memory, or UNPREDICTABLE data will be fetched. When accessing an element containing 64-bit floating point values, the Buffer Starting Address and Source Element Offset values must add to a 64-bit aligned address, and BufferPitch must be a multiple of 64-bits. VBs can only be allocated in linear (not tiled) graphics memory. As computed index values are, by definition, interpreted as unsigned values, there is no issue with accesses to locations before (lower address value) the start of the buffer. However, these wrapped indices are subject to Max Index checking (see below). 									
3	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center; vertical-align: top;">31:0</td> <td> <p>Buffer Size</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>U32 Count of bytes</td> </tr> </table> <p>This field specifies the size of the buffer in bytes. Vertex element accesses which straddle or go past the end of the buffer will return 0's for all elements. Note that BufferSize=0 indicates that there is no valid data in the buffer.</p> <table border="1" style="width: 100%;"> <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, FFFFFFFFh]</td> <td></td> </tr> </tbody> </table> </td> </tr> </table>	31:0	<p>Buffer Size</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>U32 Count of bytes</td> </tr> </table> <p>This field specifies the size of the buffer in bytes. Vertex element accesses which straddle or go past the end of the buffer will return 0's for all elements. Note that BufferSize=0 indicates that there is no valid data in the buffer.</p> <table border="1" style="width: 100%;"> <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, FFFFFFFFh]</td> <td></td> </tr> </tbody> </table>	Format:	U32 Count of bytes	Value	Name	[0, FFFFFFFFh]	
31:0	<p>Buffer Size</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td>U32 Count of bytes</td> </tr> </table> <p>This field specifies the size of the buffer in bytes. Vertex element accesses which straddle or go past the end of the buffer will return 0's for all elements. Note that BufferSize=0 indicates that there is no valid data in the buffer.</p> <table border="1" style="width: 100%;"> <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, 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]									

VERTEX_ELEMENT_STATE

VERTEX_ELEMENT_STATE										
Source:	RenderCS									
Size (in bits):	64									
Default Value:	0x00000000, 0x00000000									
Description										
<p>This structure is used in 3DSTATE_VERTEX_ELEMENTS to set the state associated with a vertex element. A vertex element is defined as an entity supplying from one to four DWord vertex components, to be stored in the vertex URB entry.</p>										
<p>The number of supported vertex elements is 34.</p>										
<p>The VF function will use this state, and possibly the state of the associated vertex buffer, to fetch/generate the source vertex element data, perform any required format conversions, padding with zeros, and store the resulting destination vertex element data into the vertex URB entry.</p>										
Programming Notes										
<ul style="list-style-type: none"> The (new) 3DSTATE_VF_SGVS command is used to specify optional insertion of VertexID and/or InstanceID into the input vertex data, logically following the processing of the VERTEX_ELEMENT_STATE structures. The VFCOMP_STORE_VID/IID encodings are no longer available in VERTEX_ELEMENT_STATE. When SourceElementFormat is set to one of the *64*_PASSTHRU formats, 64-bit components are stored in the URB without any conversion. In this case, vertex elements must be written as 128 or 256 bits, with VFCOMP_STORE_0 being used to pad the output as required. E.g., if R64_PASSTHRU is used to copy a 64-bit Red component into the URB, Component 1 must be specified as VFCOMP_STORE_0 (with Components 2,3 set to VFCOMP_NOSTORE) in order to output a 128-bit vertex element, or Components 1-3 must be specified as VFCOMP_STORE_0 in order to output a 256-bit vertex element. Likewise, use of R64G64B64_PASSTHRU requires Component 3 to be specified as VFCOMP_STORE_0 in order to output a 256-bit vertex element. When SourceElementFormat is set to one of the *64*_PASSTHRU formats then VFCOMP_STORE_SRC must be used for every valid component. Any SourceElementFormat of *64*_PASSTHRU cannot be used with an element which has edge flag enabled. 										
<p>The SourceElementFormat needs to be a single-component format with an element which has edge flag enabled.</p>										
DWord	Bit	Description								
0	31:26	<p>Vertex Buffer Index</p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U6</td> </tr> <tr> <td colspan="2">This field specifies which vertex buffer the element is sourced from.</td> </tr> <tr> <th>Value</th> <th>Name</th> </tr> <tr> <td>[0,32]</td> <td>Up to 33 VBs are supported</td> </tr> </table>	Format:	U6	This field specifies which vertex buffer the element is sourced from.		Value	Name	[0,32]	Up to 33 VBs are supported
Format:	U6									
This field specifies which vertex buffer the element is sourced from.										
Value	Name									
[0,32]	Up to 33 VBs are supported									

VERTEX_ELEMENT_STATE		
Programming Notes		
It is possible for a vertex element to include only internally-generated data (VertexID, etc.), in which case the associated vertex buffer state is ignored.		
25	Valid	
	Format:	Boolean
	Value	Name
	Description	
	1h	TRUE
	0h	FALSE
		this vertex element is used in vertex assembly
		this vertex element is not used.
24:16	Source Element Format	
	Format:	SURFACE_FORMAT
	Range: Valid formats are found in the 3D Primitive Processing FormatConversion portion of the vertex fetch chapter.	
	Format: The encoding of this field is identical the Surface Format field of the SURFACE_STATE structure, as described in the Sampler chapter.	
	This field specifies the format in which the memory-resident source data for this particular vertex element is stored in the memory buffer. This only applies to elements stored with VFCOMP_STORE_SRC component control. (All other component types have an explicit format).	
15	Edge Flag Enable	
	Format:	Enable
	Description	
	When ENABLED, the source element is interpreted as an EdgeFlag for the vertex. If the source element is zero, the EdgeFlag will be set to FALSE. If the source element is non-zero, the EdgeFlag will be set to TRUE. The EdgeFlag bit will travel down the fixed function pipeline along with the vertex handle, etc. and not be stored in the vertex data like the other vertex elements. Refer to the fixed function descriptions for how this EdgeFlag affects rendering. Edge flags are supported for the following primitive topology types only, otherwise EdgeFlagEnable must not be ENABLED.	
	<ul style="list-style-type: none"> • 3DPRIM_TRILIST* • 3DPRIM_TRISTRIP* • 3DPRIM_TRIFAN* • 3DPRIM_POLYGON 	
	If this bit is DISABLED for all valid VERTEX_ELEMENTS, the vertex will be assigned a default EdgeFlag of TRUE.	
	Edge flags are supported for all primitive topology types.	
	Programming Notes	

VERTEX_ELEMENT_STATE								
		<ul style="list-style-type: none"> This bit must only be ENABLED on the last valid VERTEX_ELEMENT structure. When set, Component 0 Control must be set to VFCOMP_STORE_SRC, and Component 1-3 Control must be set to VFCOMP_NOSTORE. 						
	14:12	Reserved Format: _____ MBZ						
	11:0	Source Element Offset Format: _____ U12 byte offset Byte offset of the source vertex element data in the structures comprising the vertex buffer. <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;">[0,2047]</td> <td></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;">See note on 64-bit float alignment in Buffer Starting Address.</td> </tr> </tbody> </table>	Value	Name	[0,2047]		Programming Notes	See note on 64-bit float alignment in Buffer Starting Address.
Value	Name							
[0,2047]								
Programming Notes								
See note on 64-bit float alignment in Buffer Starting Address.								
1	31	Reserved Format: _____ MBZ						
	30:28	Component 0 Control Format: _____ 3D_Vertex_Component_Control Refer to the 3D_Vertex_Component_Control table below						
	27	Reserved Format: _____ MBZ						
	26:24	Component 1 Control Format: _____ 3D_Vertex_Component_Control Refer to the 3D_Vertex_Component_Control table below						
	23	Reserved Format: _____ MBZ						
	22:20	Component 2 Control Format: _____ 3D_Vertex_Component_Control Refer to the 3D_Vertex_Component_Control table below						
	19	Reserved Format: _____ MBZ						
	18:16	Component 3 Control Format: _____ 3D_Vertex_Component_Control Refer to the 3D_Vertex_Component_Control table below						

VERTEX_ELEMENT_STATE		
	15:8	Reserved
		Format: MBZ
	7:0	Reserved
		Format: MBZ

Vertical Line Stride Override Message Descriptor Control Field

MDC_VLSO - Vertical Line Stride Override Message Descriptor Control Field				
Source:	BSpec			
Size (in bits):	3			
Default Value:	0x00000000			
DWord	Bit	Description		
0	2	<p>Vertical Line Stride Override</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>If set, override the Vertical Line Stride and Vertical Line Stride Offset fields in the surface state with the fields below.</p>	Format:	Enable
	Format:	Enable		
	1	<p>Vertical Line Stride</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Specifies number of lines (0 or 1) to skip between logically adjacent lines - provides support of interleaved (field) surfaces as textures.</p>	Format:	U1
Format:	U1			
0	<p>Vertical Line Stride Offset</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Specifies the offset of the initial line from the beginning of the buffer. Ignored when Override VerticalLine Stride is 0.</p>	Format:	U1	
Format:	U1			

VFE_STATE_EX

VFE_STATE_EX								
Source:	RenderCS							
Size (in bits):	256							
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000							
DWord	Bit	Description						
0	31:8	Reserved						
	7:0	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							
1	31:0	VFE Control This field is used by VFE depending on the mode of operation. See the following tables for details. If VFE Mode = AVC-IT or AVC-MC, this field is valid as defined in Table 1 13. If VFE Mode = VC1-IT, this field is valid as defined in Table 1 14. Otherwise, this field is reserved.						
2	31:0	Interface Descriptor Remap Table This field contains the interface descriptor remap table entries for the first 8 kernel indices. Each table entry has 4 bits, providing a remapping range of [0, 15]. The input of this table is the Interface Descriptor Offset within the MEDIA_OBJECT or MEDIA_OBJECT_EX command. As the table is limited to map the first 16 values, any Interface Descriptor Offset greater than 15 is not remapped. Bits 31:28: Remap for index = 7 Bits 27:24: Remap for index = 6 Bits 23:20: Remap for index = 5 Bits 19:16: Remap for index = 4 Bits 15:12: Remap for index = 3 Bits 11:8: Remap for index = 2 Bits 7:4: Remap for index = 1 Bits 3:0: Remap for index = 0						
3	31:0	Interface Descriptor Remap Table (cont) This field contains the interface descriptor remap table entries for the next 8 kernel indices (index = 8...15). Each table entry has 4 bits, providing a remapping range of [0, 15]. Bits 31:28: Remap for index = 15 Bits 27:24: Remap for index = 14 Bits 23:20: Remap for index = 13 Bits 19:16: Remap for index = 12 Bits 15:12: Remap for index = 11 Bits 11:8: Remap for index = 10 Bits 7:4: Remap for index = 9 Bits 3:0: Remap for index = 8						
4	31	Scoreboard Enable This field enables and disables the hardware scoreboard in the Media Pipeline. If this field is cleared, hardware ignores the following scoreboard state fields. This should be enabled at all times in the state and the scoreboard enable field in the MEDIA_OBJECT command should be use instead. If this field is disabled, the scratch space pointer calculation will be incorrect and any attempt to use the scoreboard later will result in a hardware hang.						
		<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Scoreboard disabled</td> </tr> <tr> <td>1</td> <td>Scoreboard enabled</td> </tr> </tbody> </table>	Value	Name	0	Scoreboard disabled	1	Scoreboard enabled
Value	Name							
0	Scoreboard disabled							
1	Scoreboard enabled							

VFE_STATE_EX									
	30	<p>Scoreboard Type</p> <p>This field selects the type of scoreboard in use.</p> <p>This field must be zero (stalling scoreboard)</p> <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>0</td> <td>Stalling Scoreboard</td> </tr> <tr> <td>1</td> <td>Reserved (for Non-stalling scoreboard)</td> </tr> </tbody> </table>	Value	Name	0	Stalling Scoreboard	1	Reserved (for Non-stalling scoreboard)	
	Value	Name							
	0	Stalling Scoreboard							
	1	Reserved (for Non-stalling scoreboard)							
	29:8	<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								
7:0	<p>Scoreboard Mask</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Boolean</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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0,7]</td> <td>Bit n</td> <td>Score n is enabled</td> </tr> </tbody> </table>	Format:	Boolean	Value	Name	Description	[0,7]	Bit n	Score n is enabled
Format:	Boolean								
Value	Name	Description							
[0,7]	Bit n	Score n is enabled							
5	31:28	<p>Scoreboard 3 Delta Y</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">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>	Format:	S3					
	Format:	S3							
	27:24	<p>Scoreboard 3 Delta X</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">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>	Format:	S3					
	Format:	S3							
	23:16	Scoreboard 2 Delta (X, Y)							
15:8	Scoreboard 1 Delta (X, Y)								
7:0	Scoreboard 0 Delta (X, Y)								
6	31:24	Scoreboard 7 Delta (X, Y)							
	23:16	Scoreboard 6 Delta (X, Y)							
	15:8	Scoreboard 5 Delta (X, Y)							
	7:0	Scoreboard 4 Delta (X, Y)							
7	31:0	<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								

VP8 Encoder StreamOut Format

VP8 Encoder StreamOut Format		
Source:	VideoCS	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:24	MbY
		Format: U8
	23:16	MbX
		Format: U8
	15:8	MbClock16
		Format: U8
	7:3	Reserved
		Format: MBZ
2	MbRcFlag	
	Format: U1	
1	MBLevelInterMBConformanceFlag	
	Format: U1	
0	MBLevelIntraMBConformanceFlag	
	Format: U1	
1	31:29	Reserved
		Format: MBZ
	28:16	MB_Residual_BitCount
		Format: U13
15:13	Reserved	
	Format: MBZ	
12:0	MB_Total_BitCount	
	Format: U13	
2	31:25	Reserved
		Format: MBZ
24:0	Cbp	
	Format: U25	
3	31	Reserved
		Format: MBZ

VP8 Encoder StreamOut Format		
	30	LastMbFlag Format: U1
	29	IntraMBFlag Format: U1
	28:24	MbType5Bits Format: U5
	23:19	Reserved Format: MBZ
	18	QindexClampHigh Format: U1
	17	QindexClampLow Format: U1
	16	CoeffClampStatus Format: U1
	15:0	Reserved Format: MBZ