

## **Intel® Iris® Xe and UHD Graphics Open Source**

### **Programmer's Reference Manual**

**For the 2020-2021 11th Generation Intel Xeon®, Core™, Celeron®,  
Pentium® Gold Processors based on the "Tiger Lake" Platform**

Volume 2d: Command Reference: Structures

December 2021, Revision 1.0



## Notices and Disclaimers

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks

Customer is responsible for safety of the overall system, including compliance with applicable safety-related requirements or standards.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document, with the sole exceptions that a) you may publish an unmodified copy and b) code included in this document is licensed subject to Zero-Clause BSD open source license (0BSD). You may create software implementations based on this document and in compliance with the foregoing that are intended to execute on the Intel product(s) referenced in this document. No rights are granted to create modifications or derivatives of this document.

The products described 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.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

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.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

## Table of Contents

3DSTATE_AMFS_BODY .....	1
3DSTATE_BINDING_TABLE_POINTERS_BODY.....	2
3DSTATE_BLEND_STATE_POINTERS_BODY.....	3
3DSTATE_CC_STATE_POINTERS_BODY .....	4
3DSTATE_CLEAR_PARAMS_BODY .....	5
3DSTATE_CLIP_BODY.....	6
3DSTATE_CONSTANT_ALL_BODY .....	12
3DSTATE_CONSTANT_ALL_DATA.....	13
3DSTATE_CONSTANT_TS_POINTER_BODY .....	14
3DSTATE_CONSTANT(Body) .....	15
3DSTATE_CPS_POINTERS_BODY .....	18
3DSTATE_DEPTH_BOUNDS_BODY .....	19
3DSTATE_DS_BODY.....	20
3DSTATE_GS_BODY.....	28
3DSTATE_HIER_DEPTH_BUFFER_BODY.....	38
3DSTATE_HS_BODY.....	41
3DSTATE_INDEX_BUFFER_BODY .....	48
3DSTATE_MULTISAMPLE_BODY.....	50
3DSTATE_PRIMITIVE_REPLICATION_BODY.....	52
3DSTATE_PS_BLEND_BODY.....	56
3DSTATE_PS_BODY .....	58
3DSTATE_PS_EXTRA_BODY .....	68
3DSTATE_PTBR_MARKER_BODY .....	74
3DSTATE_PTBR_TILE_SELECT_BODY .....	75
3DSTATE_RASTER_BODY.....	77
3DSTATE_SAMPLE_MASK_BODY.....	82
3DSTATE_SAMPLER_STATE_POINTERS_BODY.....	83
3DSTATE_SBE_BODY .....	84
3DSTATE_SBE_SWIZ_BODY .....	91
3DSTATE_SCISSOR_STATE_POINTERS_BODY.....	93
3DSTATE_SF_BODY .....	94
3DSTATE_SO_BUFFER_INDEX_BODY .....	99



<b>3DSTATE_STENCIL_BUFFER_BODY</b> .....	<b>101</b>
<b>3DSTATE_STREAMOUT_BODY</b> .....	<b>109</b>
<b>3DSTATE_TE_BODY</b> .....	<b>113</b>
<b>3DSTATE_URB_ALLOC_DS_BODY</b> .....	<b>117</b>
<b>3DSTATE_URB_ALLOC_GS_BODY</b> .....	<b>119</b>
<b>3DSTATE_URB_ALLOC_HS_BODY</b> .....	<b>121</b>
<b>3DSTATE_URB_ALLOC_VS_BODY</b> .....	<b>123</b>
<b>3DSTATE_URB_DS_BODY</b> .....	<b>125</b>
<b>3DSTATE_URB_GS_BODY</b> .....	<b>126</b>
<b>3DSTATE_URB_HS_BODY</b> .....	<b>127</b>
<b>3DSTATE_URB_VS_BODY</b> .....	<b>128</b>
<b>3DSTATE_VF_BODY</b> .....	<b>130</b>
<b>3DSTATE_VF_COMPONENT_PACKING_BODY</b> .....	<b>131</b>
<b>3DSTATE_VF_INSTANCING_BODY</b> .....	<b>132</b>
<b>3DSTATE_VF_SGVS_2_BODY</b> .....	<b>134</b>
<b>3DSTATE_VF_SGVS_BODY</b> .....	<b>137</b>
<b>3DSTATE_VF_TOPOLOGY_BODY</b> .....	<b>139</b>
<b>3DSTATE_VIEWPORT_STATE_POINTERS_CC_BODY</b> .....	<b>140</b>
<b>3DSTATE_VIEWPORT_STATE_POINTERS_SF_CLIP_BODY</b> .....	<b>141</b>
<b>3DSTATE_VS_BODY</b> .....	<b>142</b>
<b>3DSTATE_WM_BODY</b> .....	<b>150</b>
<b>3DSTATE_WM_CHROMAKEY_BODY</b> .....	<b>154</b>
<b>3DSTATE_WM_DEPTH_STENCIL_BODY</b> .....	<b>155</b>
<b>3DSTATE_WM_HZ_OP_BODY</b> .....	<b>158</b>
<b>A32 Buffer Base Address Message Header Control</b> .....	<b>164</b>
<b>A32 Scaled Header Present Message Descriptor Control Field</b> .....	<b>165</b>
<b>A32 Sideband Scale and Offset Enable Message Descriptor Control Field</b> .....	<b>166</b>
<b>A64 Data Size Message Descriptor Control Field</b> .....	<b>167</b>
<b>A64 Hword Block Message Header</b> .....	<b>168</b>
<b>A64 Hword Data Blocks Message Descriptor Control Field</b> .....	<b>169</b>
<b>A64 Oword Block Message Header</b> .....	<b>170</b>
<b>A64 Oword Data Blocks Message Descriptor Control Field</b> .....	<b>171</b>
<b>A64 Scaled Header Present Message Descriptor Control Field</b> .....	<b>172</b>
<b>AddrSubRegNum</b> .....	<b>173</b>

<b>Any Binding Table Index Message Descriptor Control Field .....</b>	<b>174</b>
<b>Atomic Float Binary Operation Message Descriptor Control Field .....</b>	<b>175</b>
<b>Atomic Float Operation Message Descriptor Control Field .....</b>	<b>176</b>
<b>Atomic Float Ternary Operation Message Descriptor Control Field.....</b>	<b>177</b>
<b>Atomic Half Float Binary Operation Message Descriptor Control Field.....</b>	<b>178</b>
<b>Atomic Integer Binary Operation Message Descriptor Control Field .....</b>	<b>179</b>
<b>Atomic Integer Operation Message Descriptor Control Field .....</b>	<b>180</b>
<b>Atomic Integer Ternary Operation Message Descriptor Control Field.....</b>	<b>181</b>
<b>Atomic Integer Unary Operation Message Descriptor Control Field.....</b>	<b>182</b>
<b>Audio Power State Format.....</b>	<b>183</b>
<b>AVC CABAC.....</b>	<b>184</b>
<b>AVC CAVLC.....</b>	<b>186</b>
<b>AVP_PAK_INSERT_OBJECT_INDIRECT_PAYLOAD .....</b>	<b>188</b>
<b>AVP_REF_LIST_ENTRY_OLD .....</b>	<b>189</b>
<b>Barrier Data Payload.....</b>	<b>190</b>
<b>BaseAddress4KByteAligned.....</b>	<b>192</b>
<b>Batch Buffer Stack Structure .....</b>	<b>193</b>
<b>Batch Stack Entry .....</b>	<b>194</b>
<b>BCS Hardware-Detected Error Bit Definitions.....</b>	<b>195</b>
<b>BINDING_TABLE_EDIT_ENTRY .....</b>	<b>196</b>
<b>Bit Definition for Interrupt Control Registers - Media.....</b>	<b>197</b>
<b>BLEND_STATE_ENTRY.....</b>	<b>199</b>
<b>BLEND_STATE .....</b>	<b>203</b>
<b>Blitter Interrupt Vector .....</b>	<b>205</b>
<b>Block Dimensions Message Header Control.....</b>	<b>206</b>
<b>Block Message Header .....</b>	<b>207</b>
<b>BR00 - BLT Opcode and Control .....</b>	<b>208</b>
<b>BR01 - Setup BLT Raster OP, Control, and Destination Offset.....</b>	<b>212</b>
<b>BR05 - Setup Expansion Background Color.....</b>	<b>215</b>
<b>BR06 - Setup Expansion Foreground Color .....</b>	<b>216</b>
<b>BR07 - Setup Blit Color Pattern Address Lower Order Address bits.....</b>	<b>217</b>
<b>BR09 - Destination Address Lower Order Address Bits.....</b>	<b>218</b>
<b>BR11 - BLT Source Pitch (Offset) .....</b>	<b>219</b>
<b>BR12 - Source Address Lower order Address bits.....</b>	<b>220</b>



<b>BR13 - BLT Raster OP, Control, and Destination Pitch</b> .....	<b>221</b>
<b>BR14 - Destination Width and Height</b> .....	<b>223</b>
<b>BR15 - Color Pattern Address Lower order Address bits</b> .....	<b>224</b>
<b>BR16 - Pattern Expansion Background and Solid Pattern Color</b> .....	<b>225</b>
<b>BR17 - Pattern Expansion Foreground Color</b> .....	<b>226</b>
<b>BR18 - Source Expansion Background and Destination Color</b> .....	<b>227</b>
<b>BR19 - Source Expansion Foreground Color</b> .....	<b>228</b>
<b>BR27 - Destination Higher Order Address</b> .....	<b>229</b>
<b>BR28 - Source Higher Order Address</b> .....	<b>230</b>
<b>BR29 - Color Pattern Higher Order Address</b> .....	<b>231</b>
<b>BR30 - Setup Blit Color Pattern Higher Order Address</b> .....	<b>232</b>
<b>Byte Masked Media Block Message Header</b> .....	<b>233</b>
<b>Byte Masked Media Block Message Header Control</b> .....	<b>234</b>
<b>CC_VIEWPORT</b> .....	<b>236</b>
<b>Channel Mask Message Descriptor Control Field</b> .....	<b>237</b>
<b>Channel Mode Message Descriptor Control Field</b> .....	<b>238</b>
<b>Clear Color</b> .....	<b>239</b>
<b>Clock Gating Disable Format</b> .....	<b>241</b>
<b>COLOR_CALC_STATE</b> .....	<b>242</b>
<b>COLOR_PROCESSING_STATE - ACE State</b> .....	<b>244</b>
<b>COLOR_PROCESSING_STATE - PROCAMP State</b> .....	<b>250</b>
<b>COLOR_PROCESSING_STATE - STD/STE State</b> .....	<b>251</b>
<b>COLOR_PROCESSING_STATE - TCC State</b> .....	<b>264</b>
<b>Color Calculator State Pointer Message Header Control</b> .....	<b>269</b>
<b>Color Code Message Header Control</b> .....	<b>270</b>
<b>COMPRESSION_PAIR_BIT</b> .....	<b>271</b>
<b>ComputeCS Hardware-Detected Error Bit Definitions</b> .....	<b>272</b>
<b>Compute Engine Interrupt Vector</b> .....	<b>273</b>
<b>Context Descriptor Format</b> .....	<b>275</b>
<b>Context Status</b> .....	<b>279</b>
<b>CPS_STATE</b> .....	<b>282</b>
<b>CSC COEFFICIENT FORMAT</b> .....	<b>286</b>
<b>Data Port 0 Message Types</b> .....	<b>287</b>
<b>Data Port 1 Message Types</b> .....	<b>288</b>

<b>Data Port Bindless Surface Extended Message Descriptor .....</b>	<b>290</b>
<b>Data Size Message Descriptor Control Field.....</b>	<b>291</b>
<b>Depth Clear Value Format.....</b>	<b>292</b>
<b>Depth Clear Value Format .....</b>	<b>293</b>
<b>DirectOperand .....</b>	<b>294</b>
<b>Display Engine Render Response Message Definition .....</b>	<b>295</b>
<b>DUALSUBSLICE_HASH_TABLE_8x8.....</b>	<b>301</b>
<b>DUALSUBSLICE_HASH_TABLE_16x8.....</b>	<b>302</b>
<b>Dword Data Payload Register .....</b>	<b>303</b>
<b>Dword SIMD8 Atomic Operation CMPWR Message Data Payload .....</b>	<b>304</b>
<b>Dword SIMD8 Data Payload .....</b>	<b>305</b>
<b>Dword SIMD16 Atomic Operation CMPWR Message Data Payload .....</b>	<b>306</b>
<b>Dword SIMD16 Data Payload .....</b>	<b>307</b>
<b>Encoder Statistics Format .....</b>	<b>308</b>
<b>EU_INSTRUCTION_BASIC_ONE_SRC.....</b>	<b>312</b>
<b>EU_INSTRUCTION_BASIC_THREE_SRC .....</b>	<b>316</b>
<b>EU_INSTRUCTION_BASIC_TWO_SRC.....</b>	<b>321</b>
<b>EU_INSTRUCTION_BRANCH_ONE_SRC.....</b>	<b>326</b>
<b>EU_INSTRUCTION_BRANCH_TWO_SRC.....</b>	<b>329</b>
<b>EU_INSTRUCTION_ILLEGAL.....</b>	<b>332</b>
<b>EU_INSTRUCTION_MATH.....</b>	<b>333</b>
<b>EU_INSTRUCTION_NOP .....</b>	<b>338</b>
<b>EU_INSTRUCTION_SEND .....</b>	<b>339</b>
<b>EU_INSTRUCTION_SYNC .....</b>	<b>344</b>
<b>Event Data Payload.....</b>	<b>347</b>
<b>ExMsgDesc.....</b>	<b>348</b>
<b>Extended Message Descriptor Render Target.....</b>	<b>349</b>
<b>Extended Message Descriptor - Sampling Engine .....</b>	<b>351</b>
<b>FFTID Message Header Control.....</b>	<b>352</b>
<b>Filter_Coefficient .....</b>	<b>353</b>
<b>Filter_Coefficients .....</b>	<b>354</b>
<b>FrameDeltaQp .....</b>	<b>355</b>
<b>FrameDeltaQpRange .....</b>	<b>356</b>
<b>Gamut_Expansion_Gamma_Correction.....</b>	<b>357</b>



<b>GGTT Entry</b> .....	<b>532</b>
<b>GraphicsAddress63-0</b> .....	<b>533</b>
<b>GraphicsAddress63-1</b> .....	<b>534</b>
<b>GraphicsAddress63-2</b> .....	<b>535</b>
<b>GraphicsAddress63-3</b> .....	<b>536</b>
<b>GraphicsAddress63-4</b> .....	<b>537</b>
<b>GraphicsAddress63-5</b> .....	<b>538</b>
<b>GraphicsAddress63-6</b> .....	<b>539</b>
<b>GraphicsAddress63-12</b> .....	<b>540</b>
<b>GTC Interrupt Bit Definition</b> .....	<b>541</b>
<b>GTPM Interrupt Vector</b> .....	<b>543</b>
<b>GUC Interrupt Vector</b> .....	<b>544</b>
<b>G-Unit Interrupt Vector</b> .....	<b>545</b>
<b>Half Precision Dual Source SIMD8 Message Data Payload Register</b> .....	<b>546</b>
<b>Half Precision OM Replicated SIMD16 Render Target Data Payload</b> .....	<b>548</b>
<b>Half Precision OM S0A SIMD8 Render Target Data Payload</b> .....	<b>549</b>
<b>Half Precision OM S0A SIMD16 Render Target Data Payload</b> .....	<b>550</b>
<b>Half Precision OM SIMD8 Dual Source Render Target Data Payload</b> .....	<b>551</b>
<b>Half Precision OM SIMD8 Render Target Data Payload</b> .....	<b>552</b>
<b>Half Precision OM SIMD16 Render Target Data Payload</b> .....	<b>553</b>
<b>Half Precision OS OM S0A SIMD8 Render Target Data Payload</b> .....	<b>554</b>
<b>Half Precision OS OM SIMD8 Dual Source Render Target Data Payload</b> .....	<b>556</b>
<b>Half Precision OS OM SIMD8 Render Target Data Payload</b> .....	<b>557</b>
<b>Half Precision OS S0A SIMD8 Render Target Data Payload</b> .....	<b>558</b>
<b>Half Precision OS SIMD8 Dual Source Render Target Data Payload</b> .....	<b>559</b>
<b>Half Precision OS SIMD8 Render Target Data Payload</b> .....	<b>560</b>
<b>Half Precision OS SZ OM S0A SIMD8 Render Target Data Payload</b> .....	<b>561</b>
<b>Half Precision OS SZ OM SIMD8 Dual Source Render Target Data Payload</b> .....	<b>563</b>
<b>Half Precision OS SZ OM SIMD8 Render Target Data Payload</b> .....	<b>565</b>
<b>Half Precision OS SZ S0A SIMD8 Render Target Data Payload</b> .....	<b>567</b>
<b>Half Precision OS SZ SIMD8 Dual Source Render Target Data Payload</b> .....	<b>569</b>
<b>Half Precision OS SZ SIMD8 Render Target Data Payload</b> .....	<b>570</b>
<b>Half Precision Replicated Pixel Render Target Data Payload Register</b> .....	<b>571</b>
<b>Half Precision Replicated SIMD16 Render Target Data Payload</b> .....	<b>572</b>



Half Precision S0A SIMD8 Render Target Data Payload .....	573
Half Precision S0A SIMD16 Render Target Data Payload .....	574
Half Precision SIMD8 Dual Source Render Target Data Payload .....	575
Half Precision SIMD8 Message Data Payload Register .....	576
Half Precision SIMD8 Render Target Data Payload .....	578
Half Precision SIMD16 Message Data Payload Register .....	579
Half Precision SIMD16 Render Target Data Payload .....	581
Half Precision SZ OM S0A SIMD8 Render Target Data Payload .....	582
Half Precision SZ OM S0A SIMD16 Render Target Data Payload .....	584
Half Precision SZ OM SIMD8 Dual Source Render Target Data Payload .....	586
Half Precision SZ OM SIMD8 Render Target Data Payload .....	587
Half Precision SZ OM SIMD16 Render Target Data Payload .....	588
Half Precision SZ S0A SIMD8 Render Target Data Payload .....	590
Half Precision SZ S0A SIMD16 Render Target Data Payload .....	591
Half Precision SZ SIMD8 Dual Source Render Target Data Payload .....	593
Half Precision SZ SIMD8 Render Target Data Payload .....	594
Half Precision SZ SIMD16 Render Target Data Payload .....	595
Hardware-Detected Error Bit Definitions .....	596
Hardware Status Page Layout .....	597
HCP_PAK_INSERT_OBJECT_INDIRECT_PAYLOAD .....	602
HCP_REF_LIST_ENTRY .....	603
HCP_TILE_POSITION_IN_CTB .....	605
HCP_TILE_POSITION_IN_CTB_MSB .....	606
HCP_WEIGHTOFFSET_CHROMA_ENTRY .....	608
HCP_WEIGHTOFFSET_CHROMA_EXT_ENTRY .....	610
HCP_WEIGHTOFFSET_LUMA_ENTRY .....	611
Header .....	612
Header Forbidden Message Descriptor Control Field .....	613
Header Present Message Descriptor Control Field .....	614
Header Required Message Descriptor Control Field .....	615
HEVC_ARBITRATION_PRIORITY .....	616
HEVC_VP9_RDOQ_LAMBDA_FIELDS .....	617
HUC_VIRTUAL_ADDR_REGION .....	618
HW Generated BINDING_TABLE_STATE .....	619



Hword 1 Block Data Payload .....	620
Hword 2 Block Data Payload .....	621
Hword 4 Block Data Payload .....	622
Hword 8 Block Data Payload .....	623
Hword Channel Mode Message Header Control.....	625
Hword Register Blocks Message Descriptor Control Field .....	626
Ignored Message Header .....	627
IndirectOperand .....	628
Inline Data Description for MFD_AVC_BSD_Object.....	629
Inline Data Description in MPEG2-IT Mode.....	637
Inline Data Description - VP8 PAK OBJECT.....	640
INTERFACE_DESCRIPTOR_DATA.....	647
INTERRUPT .....	653
Invalidate After Read Message Descriptor Control Field.....	655
JPEG .....	656
LNCF_POWER_CONTEXT .....	657
LOD Message Address Payload Control .....	659
Lower Oword Block Data Payload .....	660
LPFC_POWER_CONTEXT .....	661
LRI Data Entry.....	662
MacroOperand .....	663
Manageability Engine Interrupt Vector .....	664
MEDIA_SURFACE_STATE .....	665
MEMORY_OBJECT_CONTROL_STATE.....	674
MemoryAddressAttributes .....	675
Merged Media Block Message Header Control.....	677
Message Descriptor - Render Target Write .....	679
Message Descriptor - Sampling Engine .....	681
MFD_MPEG2_BSD_OBJECT Inline Data Description.....	684
MF_X_REFERENCE_PICTURE_BASE_ADDR.....	687
MPEG2 .....	688
MSAA Sample Number Message Address Control .....	689
MsgDesc .....	690
Named Barrier State .....	692

<b>No Event Data Payload.....</b>	<b>693</b>
<b>Normal Media Block Message Header .....</b>	<b>694</b>
<b>Normal Media Block Message Header Control .....</b>	<b>695</b>
<b>oMask Message Data Payload Register.....</b>	<b>697</b>
<b>OM Replicated SIMD16 Render Target Data Payload .....</b>	<b>699</b>
<b>OM S0A SIMD8 Render Target Data Payload.....</b>	<b>700</b>
<b>OM S0A SIMD16 Render Target Data Payload.....</b>	<b>701</b>
<b>OM SIMD8 Dual Source Render Target Data Payload.....</b>	<b>703</b>
<b>OM SIMD8 Render Target Data Payload.....</b>	<b>705</b>
<b>OM SIMD16 Render Target Data Payload .....</b>	<b>706</b>
<b>OS OM S0A SIMD8 Render Target Data Payload .....</b>	<b>708</b>
<b>OS OM SIMD8 Dual Source Render Target Data Payload.....</b>	<b>710</b>
<b>OS OM SIMD8 Render Target Data Payload.....</b>	<b>712</b>
<b>OS S0A SIMD8 Render Target Data Payload .....</b>	<b>713</b>
<b>OS SIMD8 Dual Source Render Target Data Payload .....</b>	<b>714</b>
<b>OS SIMD8 Render Target Data Payload .....</b>	<b>716</b>
<b>OS SZ OM S0A SIMD8 Render Target Data Payload .....</b>	<b>717</b>
<b>OS SZ OM SIMD8 Dual Source Render Target Data Payload .....</b>	<b>719</b>
<b>OS SZ OM SIMD8 Render Target Data Payload .....</b>	<b>721</b>
<b>OS SZ S0A SIMD8 Render Target Data Payload.....</b>	<b>722</b>
<b>OS SZ SIMD8 Dual Source Render Target Data Payload.....</b>	<b>723</b>
<b>OS SZ SIMD8 Render Target Data Payload.....</b>	<b>725</b>
<b>Oword 2 Block Data Payload .....</b>	<b>726</b>
<b>Oword 4 Block Data Payload .....</b>	<b>727</b>
<b>Oword 8 Block Data Payload .....</b>	<b>728</b>
<b>Oword A64 SIMD8 Atomic Operation CMPWR16B Message Data Payload.....</b>	<b>729</b>
<b>Oword Data Blocks Message Descriptor Control Field.....</b>	<b>731</b>
<b>Oword Data Payload Register .....</b>	<b>732</b>
<b>Oword Dual Data Blocks Message Descriptor Control Field.....</b>	<b>733</b>
<b>Performance Counter Report Format 101b .....</b>	<b>734</b>
<b>Per Thread Scratch Space Message Header Control.....</b>	<b>737</b>
<b>PIXEL_HASH_TABLE_1BIT_32ENTRY .....</b>	<b>738</b>
<b>PIXEL_HASH_TABLE_1BIT_64ENTRY .....</b>	<b>739</b>
<b>PIXEL_HASH_TABLE_1BIT_128ENTRY .....</b>	<b>740</b>



<b>PIXEL_HASH_TABLE_2BIT_64ENTRY .....</b>	<b>741</b>
<b>PIXEL_HASH_TABLE_2BIT_128ENTRY .....</b>	<b>748</b>
<b>Pixel Sample Mask Render Target Message Header Control .....</b>	<b>762</b>
<b>PPHWSP_LAYOUT .....</b>	<b>763</b>
<b>Predicate Barrier Message Data Payload .....</b>	<b>769</b>
<b>Qword Data Payload Register .....</b>	<b>770</b>
<b>Qword SIMD8 Atomic Operation CMPWR8B Message Data Payload .....</b>	<b>771</b>
<b>Qword SIMD8 Atomic Operation CMPWR Message Data Payload .....</b>	<b>772</b>
<b>Qword SIMD8 Atomic Operation Return Data Message Data Payload.....</b>	<b>773</b>
<b>Qword SIMD8 Data Payload .....</b>	<b>774</b>
<b>Qword SIMD16 Atomic Operation CMPWR8B Message Data Payload.....</b>	<b>775</b>
<b>Qword SIMD16 Atomic Operation Return Data Message Data Payload.....</b>	<b>777</b>
<b>Qword SIMD16 Data Payload .....</b>	<b>778</b>
<b>Read-Only Data Port Message Types .....</b>	<b>779</b>
<b>Read Surface Info 32-Bit Address Payload .....</b>	<b>780</b>
<b>Read Surface Info Address Payload.....</b>	<b>781</b>
<b>Read Surface Info Data Payload.....</b>	<b>782</b>
<b>RENDER_SURFACE_STATE .....</b>	<b>784</b>
<b>Render Data Port Message Types .....</b>	<b>817</b>
<b>Render Engine Interrupt Vector .....</b>	<b>818</b>
<b>Render Target Index Message Header Control .....</b>	<b>820</b>
<b>Render Target Message Header .....</b>	<b>821</b>
<b>Render Target Message Header Control.....</b>	<b>823</b>
<b>Render Target Message Header Poly .....</b>	<b>825</b>
<b>Replicated Pixel Render Target Data Payload Register .....</b>	<b>827</b>
<b>Replicated SIMD16 Render Target Data Payload .....</b>	<b>828</b>
<b>Reversed SIMD Mode 2 Message Descriptor Control Field .....</b>	<b>829</b>
<b>S0A SIMD8 Render Target Data Payload .....</b>	<b>830</b>
<b>S0A SIMD16 Render Target Data Payload .....</b>	<b>831</b>
<b>SAMPLER_BORDER_COLOR_STATE .....</b>	<b>833</b>
<b>SAMPLER_INDIRECT_STATE_BORDER_COLOR.....</b>	<b>836</b>
<b>SAMPLER_INDIRECT_STATE.....</b>	<b>838</b>
<b>SAMPLER_STATE_8x8_AVS.....</b>	<b>841</b>
<b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS .....</b>	<b>846</b>

<b>SAMPLER_STATE .....</b>	<b>850</b>
<b>Sampler Message Header .....</b>	<b>864</b>
<b>SCALER_COEFFICIENT_FORMAT .....</b>	<b>868</b>
<b>SCISSOR_RECT .....</b>	<b>869</b>
<b>Scratch Hword Block Message Header.....</b>	<b>870</b>
<b>SF_CLIP_VIEWPORT .....</b>	<b>871</b>
<b>SF_OUTPUT_ATTRIBUTE_DETAIL.....</b>	<b>873</b>
<b>SFC_8x8_AVS_COEFFICIENTS .....</b>	<b>875</b>
<b>SFC_AVS_CHROMA_COEFF_TABLE_BODY.....</b>	<b>878</b>
<b>SFC_AVS_LUMA_COEFF_TABLE_BODY.....</b>	<b>880</b>
<b>SFC_AVS_STATE_BODY .....</b>	<b>884</b>
<b>SFC_FRAME_START_BODY.....</b>	<b>887</b>
<b>SFC_HDR_STATE .....</b>	<b>888</b>
<b>SFC_HISTOGRAM_SURFACE.....</b>	<b>890</b>
<b>SFC_IEF_STATE_BODY .....</b>	<b>892</b>
<b>SFC_LOCK_BODY.....</b>	<b>904</b>
<b>SFC_STATE_BODY .....</b>	<b>906</b>
<b>SIMD1 Untyped BUFFER Surface 64-Bit Address Payload .....</b>	<b>945</b>
<b>SIMD8 Dual Source Render Target Data Payload .....</b>	<b>946</b>
<b>SIMD8 LOD Message Address Payload Control .....</b>	<b>948</b>
<b>SIMD8 Render Target Data Payload .....</b>	<b>949</b>
<b>SIMD8 Typed Surface 32-Bit Address Payload.....</b>	<b>950</b>
<b>SIMD8 Untyped BUFFER Surface 32-Bit Address Payload .....</b>	<b>951</b>
<b>SIMD8 Untyped BUFFER Surface 64-Bit Address Payload .....</b>	<b>952</b>
<b>SIMD8 Untyped SCRATCH Surface 32-Bit Address Payload.....</b>	<b>953</b>
<b>SIMD8 URB Channel Mask Message Address Payload .....</b>	<b>954</b>
<b>SIMD8 URB Offset Message Address Payload.....</b>	<b>955</b>
<b>SIMD16 Render Target Data Payload.....</b>	<b>956</b>
<b>SIMD16 Typed Surface 16-Bit Address Payload .....</b>	<b>958</b>
<b>SIMD16 Untyped BUFFER Surface 16-Bit Address Payload .....</b>	<b>959</b>
<b>SIMD16 Untyped BUFFER Surface 32-Bit Address Payload .....</b>	<b>960</b>
<b>SIMD16 Untyped BUFFER Surface 64-Bit Address Payload .....</b>	<b>961</b>
<b>SIMD16 Untyped SCRATCH Surface 16-Bit Address Payload.....</b>	<b>962</b>
<b>SIMD16 Untyped SCRATCH Surface 32-Bit Address Payload.....</b>	<b>963</b>



<b>SIMD 32-Bit Address Payload Control .....</b>	<b>964</b>
<b>SIMD 64-Bit Address Payload Control .....</b>	<b>965</b>
<b>SIMD8 32-Bit Address Payload .....</b>	<b>966</b>
<b>SIMD8 64-Bit Address Payload .....</b>	<b>967</b>
<b>SIMD16 16-Bit Address Payload .....</b>	<b>968</b>
<b>SIMD16 32-Bit Address Payload .....</b>	<b>969</b>
<b>SIMD16 64-Bit Address Payload .....</b>	<b>970</b>
<b>SIMD Mode 2 Message Descriptor Control Field .....</b>	<b>971</b>
<b>SIMD Mode 3 Message Descriptor Control Field .....</b>	<b>972</b>
<b>SLICE_HASH_TABLE.....</b>	<b>973</b>
<b>SLICE_HASHING_TABLE_ENTRY.....</b>	<b>974</b>
<b>SLM Block Message Header.....</b>	<b>975</b>
<b>Slot Group 2 Message Descriptor Control Field .....</b>	<b>976</b>
<b>Slot Group 3 Message Descriptor Control Field .....</b>	<b>977</b>
<b>Slot Group Select Render Cache Message Descriptor Control Field .....</b>	<b>978</b>
<b>SO_DECL .....</b>	<b>979</b>
<b>SO_DECL_ENTRY .....</b>	<b>981</b>
<b>Split_coding_unit_flags.....</b>	<b>982</b>
<b>SplitBaseAddress4KByteAligned.....</b>	<b>983</b>
<b>SplitBaseAddress64ByteAligned .....</b>	<b>984</b>
<b>Stateless Binding Table Index Message Descriptor Control Field.....</b>	<b>985</b>
<b>Stateless Block Message Header .....</b>	<b>986</b>
<b>Stateless Surface Message Header .....</b>	<b>987</b>
<b>Stateless Surface Pixel Mask Message Header.....</b>	<b>988</b>
<b>Stencil Message Data Payload Register .....</b>	<b>989</b>
<b>Subset Atomic Integer Trinary Operation Message Descriptor Control Field .....</b>	<b>990</b>
<b>Subset Reversed SIMD Mode 2 Message Descriptor Control Field .....</b>	<b>991</b>
<b>Subset SIMD Mode 2 Message Descriptor Control Field .....</b>	<b>992</b>
<b>Subset SIMD Mode 3 Message Descriptor Control Field .....</b>	<b>993</b>
<b>Subspan Render Target Message Header Control .....</b>	<b>994</b>
<b>Surface Binding Table Index Message Descriptor Control Field .....</b>	<b>995</b>
<b>Surface or Stateless Binding Table Index Message Descriptor Control Field.....</b>	<b>996</b>
<b>Surface Pixel Mask Message Header .....</b>	<b>997</b>
<b>SW Generated BINDING_TABLE_STATE.....</b>	<b>998</b>

<b>SWSB .....</b>	<b>999</b>
<b>SZ OM S0A SIMD8 Render Target Data Payload .....</b>	<b>1001</b>
<b>SZ OM S0A SIMD16 Render Target Data Payload .....</b>	<b>1002</b>
<b>SZ OM SIMD8 Dual Source Render Target Data Payload .....</b>	<b>1004</b>
<b>SZ OM SIMD8 Render Target Data Payload .....</b>	<b>1006</b>
<b>SZ OM SIMD16 Render Target Data Payload .....</b>	<b>1007</b>
<b>SZ S0A SIMD8 Render Target Data Payload.....</b>	<b>1009</b>
<b>SZ S0A SIMD16 Render Target Data Payload.....</b>	<b>1010</b>
<b>SZ SIMD8 Dual Source Render Target Data Payload.....</b>	<b>1012</b>
<b>SZ SIMD8 Render Target Data Payload.....</b>	<b>1014</b>
<b>SZ SIMD16 Render Target Data Payload .....</b>	<b>1015</b>
<b>Thread EOT Message Descriptor .....</b>	<b>1017</b>
<b>TILE_RECT.....</b>	<b>1018</b>
<b>Timeout Data Payload.....</b>	<b>1019</b>
<b>U24_X8 .....</b>	<b>1020</b>
<b>Untyped Write Channel Mask Message Descriptor Control Field.....</b>	<b>1021</b>
<b>Upper Oword Block Data Payload .....</b>	<b>1022</b>
<b>URB Channel Mask Payload Control.....</b>	<b>1023</b>
<b>URB Handle Message Header.....</b>	<b>1024</b>
<b>URB Handle Message Header Control.....</b>	<b>1025</b>
<b>VC1.....</b>	<b>1026</b>
<b>VCS Hardware-Detected Error Bit Definitions.....</b>	<b>1027</b>
<b>VD_CONTROL_STATE_BODY .....</b>	<b>1028</b>
<b>VDENC_64B_Aligned_Lower_Address .....</b>	<b>1029</b>
<b>VDENC_64B_Aligned_Upper_Address .....</b>	<b>1030</b>
<b>VDENC_Block_8x8_4.....</b>	<b>1031</b>
<b>VDENC_Colocated_MV_Picture.....</b>	<b>1032</b>
<b>VDENC_Delta_MV_XY.....</b>	<b>1033</b>
<b>VDENC_Down_Scaled_Reference_Picture .....</b>	<b>1035</b>
<b>VDENC_Original_Uncompressed_Picture.....</b>	<b>1036</b>
<b>VDENC_Reference_Picture .....</b>	<b>1037</b>
<b>VDENC_Row_Store_Scratch_Buffer_Picture.....</b>	<b>1038</b>
<b>VDENC_Sub_Mb_Pred_Mode .....</b>	<b>1039</b>
<b>VDENC_Surface_Control_Bits.....</b>	<b>1040</b>



<b>VDENC_Surface_State_Fields</b> .....	<b>1042</b>
<b>VEBOX_ACE_LACE_STATE</b> .....	<b>1047</b>
<b>VEBOX_ALPHA_AOI_STATE</b> .....	<b>1060</b>
<b>VEBOX_CAPTURE_PIPE_STATE</b> .....	<b>1062</b>
<b>VEBOX_CCM_STATE</b> .....	<b>1066</b>
<b>VEBOX_Ch_Dir_Filter_Coefficient</b> .....	<b>1070</b>
<b>VEBOX_CSC_STATE</b> .....	<b>1071</b>
<b>VEBOX_DNDI_STATE</b> .....	<b>1074</b>
<b>VEBOX_Filter_Coefficient</b> .....	<b>1096</b>
<b>VEBOX_FORWARD_GAMMA_CORRECTION_STATE</b> .....	<b>1097</b>
<b>VEBOX_FRONT_END_CSC_STATE</b> .....	<b>1105</b>
<b>VEBOX_GAMUT_CONTROL_STATE</b> .....	<b>1108</b>
<b>VEBOX_PROCAMP_STATE</b> .....	<b>1115</b>
<b>VEBOX_RGB_TO_GAMMA_CORRECTION</b> .....	<b>1116</b>
<b>VEBOX_STD_STE_STATE</b> .....	<b>1117</b>
<b>VEBOX_TCC_STATE</b> .....	<b>1133</b>
<b>VEBOX_VERTEX_TABLE_ENTRY</b> .....	<b>1139</b>
<b>VECS Hardware-Detected Error Bit Definitions</b> .....	<b>1140</b>
<b>VERTEX_BUFFER_STATE</b> .....	<b>1141</b>
<b>VERTEX_ELEMENT_STATE</b> .....	<b>1144</b>
<b>Vertical Line Stride Override Message Descriptor Control Field</b> .....	<b>1148</b>
<b>VideoDecoder Interrupt Vector</b> .....	<b>1149</b>
<b>VideoEnhancement Interrupt Vector</b> .....	<b>1151</b>
<b>VP8 Encoder StreamOut Format</b> .....	<b>1152</b>
<b>WDBoxOAIInterrupt Vector</b> .....	<b>1154</b>
<b>WD Interrupt Bit Definition</b> .....	<b>1155</b>
<b>Word Data Payload Register</b> .....	<b>1156</b>
<b>Word SIMD8 Atomic Operation CMPWR Message Data Payload</b> .....	<b>1158</b>
<b>Word SIMD8 Data Payload</b> .....	<b>1159</b>
<b>Word SIMD16 Atomic Operation CMPWR Message Data Payload</b> .....	<b>1160</b>
<b>Word SIMD16 Data Payload</b> .....	<b>1161</b>



## 3DSTATE\_AMFS\_BODY

3DSTATE_AMFS_BODY																
Source:	RenderCS															
Size (in bits):	32															
Default Value:	0x00000000															
DWord	Bit	Description														
0	31:6	<b>TS State Array Pointer</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>GraphicsAddress[31:6]</td></tr></table> Specifies the 64-byte aligned offset of the TS_STATE_ARRAY, with entries per texel shader. This offset is relative to the <b>Dynamic State Base Address</b> .	GraphicsAddress[31:6]													
	GraphicsAddress[31:6]															
	5	<b>Reserved</b>														
	4:3	<b>AMFS mode</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>U2</td></tr></table>	U2													
	U2															
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>normal mode <b>[Default]</b></td> <td>AMFS shades unshaded texel blocks only, marks them as shaded, no special handling of out of memory condition other than setting indicator bit in CACHE_MODE1 bit[6]</td> </tr> <tr> <td>1h</td> <td>touch mode</td> <td>forces out of memory operation, AMFS marks unshaded texel blocks as touched, texel shader dispatch is disabled</td> </tr> <tr> <td>2h</td> <td>backfill mode</td> <td>AMFS shades only touched texel blocks, touched blocks get marked as shaded, originally unshaded and shaded are left unchanged</td> </tr> <tr> <td>3h</td> <td>fallback mode</td> <td>AMFS shades unshaded texel blocks only, marks them as shaded, special handling of out of memory condition. When AMFS runs out of scratch space, texels are not shaded. Instead, they are marked as "touched". AMFS also sets indicator bit in CACHE_MODE1 bit[6]</td> </tr> </tbody> </table>	Value	Name	Description	0h	normal mode <b>[Default]</b>	AMFS shades unshaded texel blocks only, marks them as shaded, no special handling of out of memory condition other than setting indicator bit in CACHE_MODE1 bit[6]	1h	touch mode	forces out of memory operation, AMFS marks unshaded texel blocks as touched, texel shader dispatch is disabled	2h	backfill mode	AMFS shades only touched texel blocks, touched blocks get marked as shaded, originally unshaded and shaded are left unchanged	3h	fallback mode	AMFS shades unshaded texel blocks only, marks them as shaded, special handling of out of memory condition. When AMFS runs out of scratch space, texels are not shaded. Instead, they are marked as "touched". AMFS also sets indicator bit in CACHE_MODE1 bit[6]
Value	Name	Description														
0h	normal mode <b>[Default]</b>	AMFS shades unshaded texel blocks only, marks them as shaded, no special handling of out of memory condition other than setting indicator bit in CACHE_MODE1 bit[6]														
1h	touch mode	forces out of memory operation, AMFS marks unshaded texel blocks as touched, texel shader dispatch is disabled														
2h	backfill mode	AMFS shades only touched texel blocks, touched blocks get marked as shaded, originally unshaded and shaded are left unchanged														
3h	fallback mode	AMFS shades unshaded texel blocks only, marks them as shaded, special handling of out of memory condition. When AMFS runs out of scratch space, texels are not shaded. Instead, they are marked as "touched". AMFS also sets indicator bit in CACHE_MODE1 bit[6]														
		<b>Programming Notes</b> For backwards compatibility mode 0h acts the same as A0 AMFS. Mode 1h can be also used for functional validation of out of memory fallback condition														
2:0		<b>Num Valid Texel Shaders</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>U3-1</td></tr></table>	U3-1													
	U3-1															
	Specifies number of valid texel shader entries in the TS_STATE_ARRAY using n-1 encoding (1..8 valid entries).															



## 3DSTATE\_BINDING\_TABLE\_POINTERS\_BODY

3DSTATE_BINDING_TABLE_POINTERS_BODY			
Source:	RenderCS		
Size (in bits):	32		
Default Value:	0x00000000		
DWord	Bit	Description	
0	31:21	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	20:16	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	15:5	<b>Pointer to Binding Table</b>	
		Format:	SurfaceStateOffset[15:5]SW Generated BINDING_TABLE_STATE*256 When Binding Table Pool is disabled and HW Binding Table Alignment is not set to 256B alignment.
		Format:	SurfaceStateOffset[16:6]SW Generated BINDING_TABLE_STATE*256 When Binding Table Pool is enabled and HW Binding Table Alignment is not set to 256B alignment.
		Format:	SurfaceStateOffset[18:8]SW Generated BINDING_TABLE_STATE*256 When HW Binding Table Alignment is set to 256B alignment.
			Specifies an aligned address offset of the function's BINDING_TABLE_STATE. The offset's base and alignment differ depending on whether HW Binding Table is enabled and the setting of HW Binding Table Alignment field: If HW Binding Table Pool is <b>disabled</b> and the HW Binding Table Alignment is not set to 256B, the offset is relative to <b>Surface State Base Address</b> and the alignment is <b>32B</b> . If HW Binding Table Pool is <b>enabled</b> and the HW Binding Table Alignment is not set to 256B, the offset is relative to the <b>Binding Table Pool Base Address</b> and the alignment is <b>64B</b> . If HW Binding Table Pool is <b>disabled</b> and the HW Binding Table Alignment is set to 256B, the offset is relative to the <b>Surface State Base Address</b> and the alignment is <b>256B</b> . If HW Binding Table Pool is <b>enabled</b> and the HW Binding Table Alignment is set to 256B, the offset is relative to the <b>Binding Table Pool Base Address</b> and the alignment is <b>256B</b> .
	4:0	<b>Reserved</b>	
Access:		RO	
Format:		MBZ	

## 3DSTATE\_BLEND\_STATE\_POINTERS\_BODY

3DSTATE_BLEND_STATE_POINTERS_BODY					
Source:	RenderCS				
Size (in bits):	32				
Default Value:	0x00000000				
DWord	Bit	Description			
0	31:6	<p><b>Blend State Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>DynamicStateOffset[31:6]BLEND_STATE*8</td> </tr> </table> <p>Specifies the 64-byte aligned offset of the BLEND_STATE. This offset is relative to the <b>Dynamic State Base Address</b>.</p>	Format:	DynamicStateOffset[31:6]BLEND_STATE*8	
	Format:	DynamicStateOffset[31:6]BLEND_STATE*8			
	5:1	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				
0	<p><b>Blend State Pointer Valid</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit, if set, indicates that the BLEND_STATE pointer has changed and new state needs to be fetched.</p>	Format:	Enable		
Format:	Enable				



## 3DSTATE\_CC\_STATE\_POINTERS\_BODY

3DSTATE_CC_STATE_POINTERS_BODY					
Source:	RenderCS				
Size (in bits):	32				
Default Value:	0x00000000				
DWord	Bit	Description			
0	31:6	<p><b>Color Calc State Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>DynamicStateOffset[31:6]COLOR_CALC_STATE</td> </tr> </table> <p>Specifies the 64-byte aligned offset of the COLOR_CALC_STATE. This offset is relative to the <b>Dynamic State Base Address</b>.</p>	Format:	DynamicStateOffset[31:6]COLOR_CALC_STATE	
	Format:	DynamicStateOffset[31:6]COLOR_CALC_STATE			
	5:1	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				
0	<p><b>Color Calc State Pointer Valid</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If set, the hardware will fetch the CC state. This bit is context saved and restored so the CC state is considered undefined once this bit is cleared due to the possibility of the CC state changing between context switches.</p>	Format:	Enable		
Format:	Enable				

## 3DSTATE\_CLEAR\_PARAMS\_BODY

RenderCS - 3DSTATE_CLEAR_PARAMS_BODY		
Size (in bits):		64
Default Value:		0x00000000, 0x00000000
DWord	Bit	Description
0	31:0	<b>Depth Clear Value</b>
		Format: IEEE_FLOAT32
		This field defines the clear value that will be applied to the depth buffer if the Depth Buffer Clear field is enabled. It is valid only if Depth Buffer Clear Value Valid is set.
		<b>Programming Notes</b>
		The clear value must be between the min and max depth values (inclusive) defined in the CC_VIEWPORT. If the depth buffer format is D32_FLOAT, then values must be limited to the range of +0.0f and 1.0f inclusive; values outside this range are reserved.
1	31:1	<b>Reserved</b>
		Access: RO
		Format: MBZ
0	0	<b>Depth Clear Value Valid</b>
		Format: Boolean
		This field enables the <b>Depth Clear Value</b> . If clear, the depth clear value is obtained from interpolated depth of an arbitrary pixel of the primitive rendered with <b>Depth Buffer Clear</b> set in WM_STATE or 3DSTATE_WM. If set, the depth clear value is obtained from the <b>Depth Clear Value</b> field of this command.

## 3DSTATE\_CLIP\_BODY

3DSTATE_CLIP_BODY									
Source:	RenderCS								
Size (in bits):	96								
Default Value:	0x00000000, 0x00000000, 0x00000000								
DWord	Bit	Description							
0	31:21	<b>Reserved</b>							
		Access: RO							
		Format: MBZ							
	20	<b>Force User Clip Distance Cull Test Enable Bitmask</b>							
		Format: Enable							
This field provides a work around override for the computation of SOL_INT::Render_Enable									
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Normal</td> <td>Clip_INT::User Clip Distance Cull Test Enable Bitmask normally</td> </tr> <tr> <td>1h</td> <td>Force</td> <td>Forces Clip_INT::User Clip Distance Cull Test Enable Bitmask to use the value in 3DSTATE_CLIP:: User Clip Distance Cull Test Enable Bitmask</td> </tr> </tbody> </table>		Value	Name	Description	0h	Normal	Clip_INT::User Clip Distance Cull Test Enable Bitmask normally	1h	Force
Value	Name	Description							
0h	Normal	Clip_INT::User Clip Distance Cull Test Enable Bitmask normally							
1h	Force	Forces Clip_INT::User Clip Distance Cull Test Enable Bitmask to use the value in 3DSTATE_CLIP:: User Clip Distance Cull Test Enable Bitmask							
19	<b>Vertex Sub Pixel Precision Select</b>								
	Format: U1								
	Selects the number of fractional bits maintained in the vertex data								
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>8 Bit</td> <td>8 sub pixel precision bits maintained</td> </tr> <tr> <td>1h</td> <td>4 Bit</td> <td>4 sub pixel precision bits maintained</td> </tr> </tbody> </table>	Value	Name	Description	0h	8 Bit	8 sub pixel precision bits maintained	1h	4 Bit
Value	Name	Description							
0h	8 Bit	8 sub pixel precision bits maintained							
1h	4 Bit	4 sub pixel precision bits maintained							
18	<b>Early Cull Enable</b>								
	Format: Enable								
	This field is used to enable/disable the EarlyCull function. When this bit is set triangles are checked if they are backface culled before proceeding through must clip function.								
	<p style="text-align: center;"><b>Programming Notes</b></p> Setting this bit must not impact functionality, this state only controls the performance of the must clip function. Vertex Sub Pixel Precision Select precision must be set to "8 bit" in order avoid precision issues.								
17	<b>Force User Clip Distance Clip Test Enable Bitmask</b>								
	Format: Enable								
	This field provides a work around override for the computation of SOL_INT::Render_Enable.								
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Normal</td> <td>Clip_INT:: User Clip Distance Clip Test Enable Bitmask normally</td> </tr> <tr> <td>1b</td> <td>Force</td> <td>Forces Clip_INT:: User Clip Distance Clip Test Enable Bitmask to use the value in 3DSTATE_CLIP::User Clip Distance Clip Test Enable Bitmask</td> </tr> </tbody> </table>	Value	Name	Description	0b	Normal	Clip_INT:: User Clip Distance Clip Test Enable Bitmask normally	1b	Force
Value	Name	Description							
0b	Normal	Clip_INT:: User Clip Distance Clip Test Enable Bitmask normally							
1b	Force	Forces Clip_INT:: User Clip Distance Clip Test Enable Bitmask to use the value in 3DSTATE_CLIP::User Clip Distance Clip Test Enable Bitmask							

## 3DSTATE\_CLIP\_BODY

	16	<b>Force Clip Mode</b>	Format: <span style="float: right;">Enable</span>													
	This field provides a work around override for the computation of SOL_INT::Render_Enable.															
	<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>0b</td> <td>Normal</td> <td>Clip_INT::Clip Mode is computed normally.</td> </tr> <tr> <td>1b</td> <td>Force</td> <td>Forces Clip_INT::Clip Mode to use the value in 3DSTATE_CLIP::User Clip Mode.</td> </tr> </tbody> </table>					Value	Name	Description	0b	Normal	Clip_INT::Clip Mode is computed normally.	1b	Force	Forces Clip_INT::Clip Mode to use the value in 3DSTATE_CLIP::User Clip Mode.		
	Value	Name	Description													
	0b	Normal	Clip_INT::Clip Mode is computed normally.													
	1b	Force	Forces Clip_INT::Clip Mode to use the value in 3DSTATE_CLIP::User Clip Mode.													
	15:12	<b>Reserved</b>	Access: <span style="float: right;">RO</span>													
	Format: <span style="float: right;">MBZ</span>															
	11:10	<b>Clipper Statistics Enable</b>	This bit controls whether Clip-unit-specific statistics register(s) can be incremented.													
	<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>00h</td> <td>Disable</td> <td>CL_INVOCATIONS_COUNT cannot increment</td> </tr> <tr> <td>01h</td> <td>Increment by one</td> <td>CL_INVOCATIONS_COUNT can increment</td> </tr> <tr> <td>03h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>					Value	Name	Description	00h	Disable	CL_INVOCATIONS_COUNT cannot increment	01h	Increment by one	CL_INVOCATIONS_COUNT can increment	03h	Reserved
Value	Name	Description														
00h	Disable	CL_INVOCATIONS_COUNT cannot increment														
01h	Increment by one	CL_INVOCATIONS_COUNT can increment														
03h	Reserved															
9	<b>Reserved</b>	Access: <span style="float: right;">RO</span>														
Format: <span style="float: right;">MBZ</span>																
8	<b>Reserved</b>	Access: <span style="float: right;">RO</span>														
Format: <span style="float: right;">MBZ</span>																
7:0	<b>User Clip Distance Cull Test Enable Bitmask</b>	Format: <span style="float: right;">Enable[8]</span>														
This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept determination needs to be made (does not cause a must clip).DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.																
1	31	<b>Clip Enable</b>	Format: <span style="float: right;">Enable</span>													
	Specifies whether the Clip function is enabled or disabled (pass-through).															
	30	<b>API Mode</b>	Controls the definition of the NEAR clipping plane													
	<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>0h</td> <td>OGL</td> <td>NEAR VP boundary == 0.0 (NDC)</td> </tr> </tbody> </table>					Value	Name	Description	0h	OGL	NEAR VP boundary == 0.0 (NDC)					
Value	Name	Description														
0h	OGL	NEAR VP boundary == 0.0 (NDC)														
29	<b>Reserved</b>	Access: <span style="float: right;">RO</span>														
Format: <span style="float: right;">MBZ</span>																

## 3DSTATE\_CLIP\_BODY

28	<b>Viewport XY Clip Test Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field is used to control whether the Viewport X, Y extents [-1,1] are considered in VertexClipTest. If both the Guardband and Viewport XY ClipTest are DISABLED, all vertices are considered "visible" with respect to the XY directions.</p>		Format:	Enable																			
Format:	Enable																						
27	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ																	
Access:	RO																						
Format:	MBZ																						
26	<b>Guardband Clip Test Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field is used to control whether the Guardband X, Y extents are considered in VertexClipTest for non-point objects. If the Guardband ClipTest is DISABLED but the Viewport XY ClipTest is ENABLED, ClipDetermination operates as if the Guardband were coincident with the Viewport. If both the Guardband and Viewport XY ClipTest are DISABLED, all vertices are considered "visible" with respect to the XY directions.</p>		Format:	Enable																			
Format:	Enable																						
25:24	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ																	
Access:	RO																						
Format:	MBZ																						
23:16	<b>User Clip Distance Clip Test Enable Bitmask</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable[8]</td> </tr> </table> <p>This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept / must clip determination needs to be made. DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.</p>		Format:	Enable[8]																			
Format:	Enable[8]																						
15:13	<b>Clip Mode</b> <p>This field specifies a general mode of the CLIP unit, when the CLIP unit is ENABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="text-align: left;">Value</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NORMAL</td> <td>TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>2h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>3h</td> <td>REJECT_ALL</td> <td>All objects are discarded</td> </tr> <tr> <td>4h</td> <td>ACCEPT_ALL</td> <td>All objects (except BAD objects) are trivially accepted. This effectively disables the clip-test/clip-determination function. Note that the CLIP unit will still filter out adjacency information, which may be required since the SF unit does not accept primitives with adjacency.</td> </tr> <tr> <td>5h-7h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>		Value	Name	Description	0h	NORMAL	TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded	1h	Reserved		2h	Reserved		3h	REJECT_ALL	All objects are discarded	4h	ACCEPT_ALL	All objects (except BAD objects) are trivially accepted. This effectively disables the clip-test/clip-determination function. Note that the CLIP unit will still filter out adjacency information, which may be required since the SF unit does not accept primitives with adjacency.	5h-7h	Reserved	
Value	Name	Description																					
0h	NORMAL	TrivialAccept objects are passed down the pipeline, MustClip objects Clipped in the Fixed Function Clipper HW, TrivialReject and BAD objects are discarded																					
1h	Reserved																						
2h	Reserved																						
3h	REJECT_ALL	All objects are discarded																					
4h	ACCEPT_ALL	All objects (except BAD objects) are trivially accepted. This effectively disables the clip-test/clip-determination function. Note that the CLIP unit will still filter out adjacency information, which may be required since the SF unit does not accept primitives with adjacency.																					
5h-7h	Reserved																						



<b>3DSTATE_CLIP_BODY</b>								
12:10	<b>Reserved</b>							
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO							
Format:	MBZ							
9	<b>Perspective Divide Disable</b>							
	<table border="1"> <tr> <td>Format:</td> <td>Disable</td> </tr> </table> <p>This field disables the Perspective Divide function performed on homogeneous position read from the URB. This feature can be used by software to submit pre-transformed "screen-space" geometry for rasterization. This likely requires the W component of positions to contain "rhw" (aka 1/w) in order to support perspective-correct interpolation of vertex attributes. Likewise, the X, Y, Z components will likely be required to be X/W, Y/W, Z/W. Note that the device does not support clipping when perspective divide is disabled. Software must specify CLIPMODE_ACCEPT_ALL whenever it disables perspective divide. This implies that software must ensure that object positions are completely contained within the "guardband" screen-space limits imposed by the SF unit (e.g., by clipping in CPU SW before submitting the objects).</p>	Format:	Disable					
Format:	Disable							
8	<b>Non-Perspective Barycentric Enable</b>							
	<table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This field enables computation of non-perspective barycentric parameters in the clipper, which are sent to SF unit in the must clip case. This field must be enabled if any non-perspective interpolation modes are used in pixel shader.</p>	Format:	Enable					
	Format:	Enable						
	<b>Programming Notes</b>							
<p>This field must be set whenever Enable bits 3 or 4 or 5 of 3DSTATE_WM:Barycentric Interpolation Mode is set. This indicates that one of the Non-perspective barycentric interpolation modes are used.</p> <p>This field must be set if the 3DSTATE_PS_EXTRA:Pixel Shader Requires Non-Perspective Bary Plane Coefficients is set.</p>								
7:6	<b>Reserved</b>							
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO							
Format:	MBZ							
5:4	<b>Triangle Strip/List Provoking Vertex Select</b>							
	<table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table>	Format:	U2					
	Format:	U2						
	enumerated type							
	This field selects which vertex of a triangle (in a triangle strip or list primitive) is considered the "provoking vertex".							
<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>0</td> </tr> <tr> <td>1h</td> <td>1</td> </tr> <tr> <td>2h</td> <td>2</td> </tr> </tbody> </table>	Value	Name	0h	0	1h	1	2h	2
Value	Name							
0h	0							
1h	1							
2h	2							

<b>3DSTATE_CLIP_BODY</b>		
	3h	Reserved
3:2	<b>Line Strip/List Provoking Vertex Select</b>	
	Format:	U2
	enumerated type	
	This field selects which vertex of a line (in a line strip or list primitive) is considered the "provoking vertex".	
	Value	Name
	0h	0
	1h	1
	2h	Reserved
	3h	Reserved
	1:0	<b>Triangle Fan Provoking Vertex Select</b>
Format:		U2
enumerated type		
This field selects which vertex of a triangle (in a triangle fan primitive) is considered the "provoking vertex".		
Value		Name
0h		0
1h		1
2h		2
3h		Reserved
2		<b>Reserved</b>
	Access:	RO
	Format:	MBZ
	<b>27:17 Minimum Point Width</b>	
	Format:	U8.3
	This value is used to clamp read-back PointWidth values.	
	<b>16:6 Maximum Point Width</b>	
	Format:	U8.3
	This value is used to clamp read-back PointWidth values.	
	<b>5 Force Zero RTA Index Enable</b>	
Format:	Enable	
If set, the Clip unit will ignore the read-back RTAIndex and operate as if the value 0 was read-		

<b>3DSTATE_CLIP_BODY</b>					
	back. If clear, the read-back value is used.				
4	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				
3:0	<b>Maximum VP Index</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U4-1</td> </tr> </table> <p>This field specifies the maximum valid VPIndex value, corresponding to the number of active viewports. If the source of the VPIndex exceeds this maximum value, a VPIndex value of 0 is passed down the pipeline. Note that this clamping does not affect a VPIndex value stored in the URB.</p>	Format:	U4-1		
Format:	U4-1				



## 3DSTATE\_CONSTANT\_ALL\_BODY

3DSTATE_CONSTANT_ALL_BODY		
Source:	RenderCS	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	255:0	<b>Constant Body</b> Format: <b>3DSTATE_CONSTANT_ALL_DATA[4]</b>

## 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	<b>Pointer To Constant Buffer</b> Format: VIRTUAL_ADDR[63:5] The value of this field is the virtual address of the location of the push constant buffer. <b>Programming Notes</b> Constant buffers must be allocated in linear (not tiled) graphics memory.
		<b>Constant Buffer Read Length</b> Format: U5 This field specifies the length of the constant data to be loaded from memory in 256-bit units. <b>Programming Notes</b> <ul style="list-style-type: none"> <li>The sum of the read length fields for all pointers must be less than or equal to the size of 64</li> <li>Zero means there no data to fetch for this buffer pointer.</li> </ul>

## 3DSTATE\_CONSTANT\_TS\_POINTER\_BODY

3DSTATE_CONSTANT_TS_POINTER_BODY				
Source:	RenderCS			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0..1	63:0	<p><b>Constant Body</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3DSTATE_CONSTANT_ALL_DATA</b></td> </tr> </table> <p>Specifies the 64-byte aligned graphics address and length of constant data to be pushed as Texel Shader payload. The push constant payload and its length is common to all Texel Shader slots.</p>	Format:	<b>3DSTATE_CONSTANT_ALL_DATA</b>
Format:	<b>3DSTATE_CONSTANT_ALL_DATA</b>			

## 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	<b>Constant Buffer 1 Read Length</b>
		Format: U16
		This field specifies the length of the constant data to be loaded from memory in 256-bit units.
		<p style="text-align: center;"><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>The sum of all four read length fields must be less than or equal to the size of 64</li> <li>Setting the value of the register to zero will disable buffer 1.</li> <li>If disabled, the <b>Pointer to Constant Buffer 1</b> must be programmed to zero.</li> </ul>
15:0	15:0	<b>Constant Buffer 0 Read Length</b>
		Format: U16
		This field specifies the length of the constant data to be loaded from memory in 256-bit units.
		<p style="text-align: center;"><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>The sum of all four read length fields must be less than or equal to the size of 64</li> <li>Setting the value of the register to zero will disable buffer 0.</li> <li>If disabled, the <b>Pointer to Constant Buffer 0</b> must be programmed to zero.</li> </ul>
1	31:16	<b>Constant Buffer 3 Read Length</b>
		Format: U16
		This field specifies the length of the constant data to be loaded from memory in 256-bit units.
		<p style="text-align: center;"><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>The sum of all four read length fields must be less than or equal to the size of 64</li> <li>Setting the value of the register to zero will disable buffer 3.</li> <li>If disabled, the <b>Pointer to Constant Buffer 3</b> must be programmed to zero.</li> </ul>
15:0	15:0	<b>Constant Buffer 2 Read Length</b>
		Format: U16
		This field specifies the length of the constant data to be loaded from memory in 256-bit units.
		<p style="text-align: center;"><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>The sum of all four read length fields must be less than or equal to the size of 64</li> <li>Setting the value of the register to zero will disable buffer 2.</li> </ul>

<b>3DSTATE_CONSTANT(Body)</b>		
		<ul style="list-style-type: none"> <li>If disabled, the <b>Pointer to Constant Buffer 2</b> must be programmed to zero.</li> </ul>
2..3	63:5	<b>Pointer To Constant Buffer 0</b> Format: VIRTUAL_ADDR[63:5]  <div style="text-align: center;"><b>Description</b></div> The value of this field is the virtual address of the location of the push constant buffer 0. GraphicsAddress [63:48] are ignored by the HW and assumed to be in correct canonical form [63:48] == [47].  <div style="text-align: center;"><b>Programming Notes</b></div> Constant buffers must be allocated in linear (not tiled) graphics memory.
	4:0	<b>Reserved</b> Access: RO Format: MBZ
4..5	63:5	<b>Pointer To Constant Buffer 1</b> Format: VIRTUAL_ADDR[63:5]  This field points to the location of Constant Buffer 1. If gather constants are enabled This field is an offset of constant Buffer1 from the Gather Pool BASE ADDRESS. 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].  <div style="text-align: center;"><b>Programming Notes</b></div> Constant buffers must be allocated in linear (not tiled) graphics memory.
	4:0	<b>Reserved</b> Access: RO Format: MBZ
6..7	63:5	<b>Pointer To Constant Buffer 2</b> Format: VIRTUAL_ADDR[63:5] 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].  <div style="text-align: center;"><b>Programming Notes</b></div> Constant buffers must be allocated in linear (not tiled) graphics memory.
	4:0	<b>Reserved</b> Access: RO



<b>3DSTATE_CONSTANT(Body)</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			
8..9	63:5	<b>Pointer To Constant Buffer 3</b>		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>VIRTUAL_ADDR[63:5]</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>	Format:	VIRTUAL_ADDR[63:5]
		Format:	VIRTUAL_ADDR[63:5]	
	<b>Programming Notes</b>			
Constant buffers must be allocated in linear (not tiled) graphics memory.				
	4:0	<b>Reserved</b>		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> </table>	Access:	RO
Access:	RO			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ
Format:	MBZ			



## 3DSTATE\_CPS\_POINTERS\_BODY

3DSTATE_CPS_POINTERS_BODY					
Source:	RenderCS				
Size (in bits):	32				
Default Value:	0x00000000				
DWord	Bit	Description			
0	31:5	<p><b>Coarse Pixel Shading State Array Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>DynamicStateOffset[31:5]</td> </tr> </table> <p>Specifies the 32-byte aligned address offset of the array of CPS_STATE states. Each CPS_STATE in the array corresponds to a Viewport index in the range [0..15]. SW must program all the CPS_STATES in the array corresponding to valid Viewport indices.</p> <p>This offset is relative to the Dynamic State Base Address.</p> <p>When 3DSTATE_PS:Pixel Shader Is Per Coarse Pixel is not set, HW does not fetch or depend on any CPS pointers to be valid.</p> <p>The hardware might not order pixels across viewports if multiple CPS_STATE entries map any(x,y) to different CPsizes.</p>	Format:	DynamicStateOffset[31:5]	
	Format:	DynamicStateOffset[31:5]			
4:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				

## 3DSTATE\_DEPTH\_BOUNDS\_BODY

3DSTATE_DEPTH_BOUNDS_BODY																
Source:	RenderCS															
Size (in bits):	96															
Default Value:	0x00000000, 0x00000000, 0x00000000															
DWord	Bit	Description														
0	31:1	<b>Reserved</b>														
		Access: RO														
		Format: MBZ														
	0	<b>Depth Bounds Test Enable</b> Enables the depth bounds test <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>Depth Bounds test is disabled.</td> </tr> <tr> <td>1</td> <td>Enabled</td> <td>Depth Bounds test is enabled. If (Z Min Value) &lt;= (Destination Z Value) &lt;= (Z Max Value) the depth bounds test passes. Otherwise the depth bounds test fails and the sample is discarded.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled</td> </tr> <tr> <td>1</td> <td>Enabled</td> </tr> </tbody> </table>	Value	Name	Description	0	Disabled	Depth Bounds test is disabled.	1	Enabled	Depth Bounds test is enabled. If (Z Min Value) <= (Destination Z Value) <= (Z Max Value) the depth bounds test passes. Otherwise the depth bounds test fails and the sample is discarded.	Value	Name	0	Disabled	1
Value	Name	Description														
0	Disabled	Depth Bounds test is disabled.														
1	Enabled	Depth Bounds test is enabled. If (Z Min Value) <= (Destination Z Value) <= (Z Max Value) the depth bounds test passes. Otherwise the depth bounds test fails and the sample is discarded.														
Value	Name															
0	Disabled															
1	Enabled															
1	31:0	<b>Depth Bounds Test Min Value</b>														
		Format: IEEE_FLOAT This field specifies the minimum Z value to be used in the depth bounds test. This value should be in 32-bit Float. HW will clamp to min value of +0 if set to below +0.														
2	31:0	<b>Depth Bounds Test Max Value</b>														
		Format: IEEE_FLOAT This field specifies the maximum Z value to be used in the depth bounds test. This value should be in 32-bit Float. HW will clamp to max value of +1 if set to greater than +1.														

## 3DSTATE\_DS\_BODY

3DSTATE_DS_BODY												
Source:	RenderCS											
Size (in bits):	320											
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000											
DWord	Bit	Description										
0..1	63:6	<p><b>Kernel Start Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[63:6]</td> </tr> </table> <p>This field specifies the starting location of the kernel program run by threads spawned by this FF unit. It is specified as a 64-byte-granular offset from the Instruction Base Address. This field is ignored if DS Function Enable is DISABLED.</p>	Format:	InstructionBaseOffset[63:6]								
	Format:	InstructionBaseOffset[63:6]										
5:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ							
Access:	RO											
Format:	MBZ											
2	31	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
		Access:	RO									
	Format:	MBZ										
	30	<p><b>Vector Mask Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Upon subsequent DS thread dispatches, this bit is loaded into the EU's Vector Mask Enable (VME, cr0.0[3]) thread state. Refer to EU documentation for the definition and use of VME state.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Dmask</td> <td>The EU will use the Dispatch Mask (supplied by the DS stage) for instruction execution.</td> </tr> <tr> <td>1h</td> <td>Vmask</td> <td>The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>Under normal conditions SW shall specify DMask, as the DS stage will provide a Dispatch Mask appropriate to SIMD4x2 or SIMD8 thread execution (as a function of dispatch mode). E.g., for SIMD4x2 thread execution, the DS stage will generate a Dispatch Mask that is equal to what the EU would use as the Vector Mask. For SIMD8 execution there is no known usage model for use of Vector Mask (as there is for PS shaders).</p>	Format:	Enable	Value	Name	Description	0h	Dmask	The EU will use the Dispatch Mask (supplied by the DS stage) for instruction execution.	1h	Vmask
Format:	Enable											
Value	Name	Description										
0h	Dmask	The EU will use the Dispatch Mask (supplied by the DS stage) for instruction execution.										
1h	Vmask	The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.										
29:27	<p><b>Sampler Count</b></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. This field is ignored if DS Function Enable is DISABLED.</p>	Format:	U3									
Format:	U3											

## 3DSTATE\_DS\_BODY

	Value	Name	Description
	0h	No Samplers	No samplers used
	1h	1-4 Samplers	between 1 and 4 samplers used
	2h	5-8 Samplers	between 5 and 8 samplers used
	3h	9-12 Samplers	between 9 and 12 samplers used
	4h	13-16 Samplers	between 13 and 16 samplers used
26	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
25:18	<b>Binding Table Entry Count</b>		
	Format:	U8	
	<p>When HW Generated Binding Table is disabled: Specifies how many binding table entries the kernel uses. Used only for prefetching of the binding table entries and associated surface state.  <b>Note:</b> For kernels using a large number of binding table entries, it may be wise to set this field to zero to avoid prefetching too many entries and thrashing the state cache. This field is ignored if DS Function Enable is DISABLED.</p> <p>When HW Generated Binding Table bit is enabled: This field indicates which cache lines (512bit units - 32 Binding Table Entry section) should be fetched. Each bit in this field corresponds to a cache line. Only the 1st 4 non-zero Binding Table entries of each 32 Binding Table entry section prefetched will have its surface state prefetched.</p>		
	<b>Value</b>	<b>Name</b>	
	[0,255]		
	<b>Programming Notes</b>		
	When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.		
17	<b>Thread Dispatch Priority</b>		
	Format:	U1	
	Specifies the priority of the thread for dispatch: This field is ignored if DS Function Enable is DISABLED.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	Normal	Normal Priority
	1h	High	High Priority
16	<b>Floating Point Mode</b>		
	Format:	U1	
	Specifies the initial floating point mode used by the dispatched thread. This field is ignored if DS Function Enable is DISABLED.		

## 3DSTATE\_DS\_BODY

		Value	Name	Description
		0h	IEEE-754	Use IEEE-754 Rules
		1h	Alternate	Use alternate rules
	15	<b>Reserved</b>		
		Access:		RO
		Format:		MBZ
	14	<b>Accesses UAV</b>		
		Format:		Enable
	This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.			
	<b>Programming Notes</b>			
	This field must not be set when DS Function Enable is disabled.			
	13	<b>Illegal Opcode Exception Enable</b>		
		Format:		Enable
	This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment. This field is ignored if DS Function Enable is DISABLED.			
	12:11	<b>Reserved</b>		
		Access:		RO
		Format:		MBZ
	10:8	<b>Reserved</b>		
	Format:		MBZ	
7	<b>Software Exception Enable</b>			
	Format:		Enable	
This bit gets loaded into EU CR0.1[13] (note the bit # difference). See Exceptions and ISA Execution Environment. This field is ignored if DS Function Enable is DISABLED.				
6:0	<b>Reserved</b>			
	Access:		RO	
	Format:		MBZ	
3..4	63:32	<b>Reserved</b>		
		Access:		RO
		Format:		MBZ
	31:10	<b>Scratch Space Base Pointer</b>		
		Format:	GeneralStateOffset[31:10]	
Specifies the starting location of the scratch space area allocated to this FF unit as a 1K-byte aligned offset from the General State Base Address. If required, each thread spawned by this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space. The				

<b>3DSTATE_DS_BODY</b>										
		<p>computed offset of the thread-specific portion will be passed in the thread payload as Scratch Space Offset. The thread is expected to utilize "stateless" DataPort read/write requests to access scratch space, where the DataPort will cause the General State Base Address to be added to the offset passed in the request header. This field is ignored if DS Function Enable is DISABLED.</p>								
	9:4	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO									
Format:	MBZ									
	3:0	<p><b>Per-Thread Scratch Space</b></p> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Specifies the amount of scratch space to be allocated to each thread spawned by this FF unit. The driver must allocate enough contiguous scratch space, starting at the Scratch Space Base Pointer, to ensure that the Maximum Number of Threads can each get Per-Thread Scratch Space size without exceeding the driver-allocated scratch space. This field is ignored if DS Function Enable is DISABLED.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td>indicating [1K Bytes, 2M Bytes]</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This amount is available to the kernel for information only. It will be passed verbatim (if not altered by the kernel) to the Data Port in any scratch space access messages, but the Data Port will ignore it.</p>	Format:	U4	Value	Name	[0,11]	indicating [1K Bytes, 2M Bytes]		
Format:	U4									
Value	Name									
[0,11]	indicating [1K Bytes, 2M Bytes]									
5	31:25	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO									
Format:	MBZ									
	24:20	<p><b>Dispatch GRF Start Register For URB Data</b></p> <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p style="text-align: center;"><b>Description</b></p> <p>Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. This field is ignored if DS Function Enable is DISABLED. When SIMD8_SINGLE_OR_DUAL_PATCH dispatch mode is selected, HW shall increment the GRF start register by 1 when a dual patch simd8 thread is dispatched AND 3DSTATE_DS::PrimitiveIDNotRequired is not set.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td></td> <td>indicating GRF [R0, R31]</td> </tr> </tbody> </table>	Format:	U5	Value	Name	Description	[0,31]		indicating GRF [R0, R31]
Format:	U5									
Value	Name	Description								
[0,31]		indicating GRF [R0, R31]								
	19:18	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO									
Format:	MBZ									

## 3DSTATE\_DS\_BODY

	17:11	<b>Patch URB Entry Read Length</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 65%;">Format:</td> <td style="width: 35%; text-align: center;">U7</td> </tr> </table> <p>Specifies how much data (in 256-bit units) is to be read from the Patch URB entry and passed in the DS thread payload. This field is ignored if DS Function Enable is DISABLED.</p> <table border="1" 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;">[0,64]</td> <td></td> </tr> </tbody> </table>	Format:	U7	Value	Name	[0,64]			
Format:	U7										
Value	Name										
[0,64]											
	10	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 65%;">Access:</td> <td style="width: 35%; text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO										
Format:	MBZ										
	9:4	<b>Patch URB Entry Read Offset</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 65%;">Format:</td> <td style="width: 35%; text-align: center;">U6</td> </tr> </table> <p>Specifies the offset (in 256-bit units) at which Patch URB data is to be read from the URB before being included in the thread payload. This field is ignored if DS Function Enable is DISABLED.</p> <table border="1" 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;">[0,63]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,63]			
Format:	U6										
Value	Name										
[0,63]											
	3:0	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 65%;">Access:</td> <td style="width: 35%; text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO										
Format:	MBZ										
6	31	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 65%;">Access:</td> <td style="width: 35%; text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO										
Format:	MBZ										
	30:21	<b>Maximum Number of Threads</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 65%;">Format:</td> <td style="width: 35%; text-align: center;">U10-1</td> </tr> </table> <p>Specifies the maximum number of simultaneous DS threads allowed to be active. Used to avoid using up the scratch space. Programming the value of the max threads over the number of threads based off number of threads supported in the execution units may improve performance since the architecture allows threads to be buffered between the check for max threads and the actual dispatch into the EU. Programming the max values to a number less than the number of threads supported in the execution units may reduce performance. This field is ignored if DS Function Enable is DISABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%; text-align: center;">Value</th> <th style="width: 25%; text-align: center;">Name</th> <th style="width: 50%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,545]</td> <td></td> <td>indicating thread count of [1,546]</td> </tr> </tbody> </table>	Format:	U10-1	Value	Name	Description	[0,545]		indicating thread count of [1,546]
Format:	U10-1										
Value	Name	Description									
[0,545]		indicating thread count of [1,546]									
	20:11	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 65%;">Access:</td> <td style="width: 35%; text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO										
Format:	MBZ										
	10	<b>Statistics Enable</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 65%;">Format:</td> <td style="width: 35%; text-align: center;">Enable</td> </tr> </table> <p>If ENABLED, this FF unit will engage in statistics gathering. Refer to the Statistics Gathering section.</p>	Format:	Enable						
Format:	Enable										



<b>3DSTATE_DS_BODY</b>				
	<p>If DISABLED, statistics information associated with this FF stage will be left unchanged. This field is ignored if DS Function Enable is DISABLED.</p>			
9	<b>PrimitiveID Not Required</b>			
	Format:	Boolean		
<b>Description</b>				
<p>Software shall set this bit whenever the active DS kernel(s) do not require PrimitiveID as input. When this bit is set, (a) the R1 PrimitiveID phase will not be included in the thread payload (DUAL_PATCH) and (b) the PrimitiveID field in the R0 payload (SINGLE_PATCH) will become UNDEFINED.</p>				
8:5	<b>Reserved</b>			
	Access:	RO		
	Format:	MBZ		
4:3	<b>Dispatch Mode</b>			
	Format:	U2		
	<p>This field specifies how the DS stage generates DS thread requests, and correspondingly impacts the DS thread payload. The setting of this field must agree with how the DS kernel was compiled. This field is ignored if DS Function Enable is DISABLED.</p>			
	<b>Value</b>	<b>Name</b>	<b>Description</b>	<b>Programming Notes</b>
	1h	SIMD8_SINGLE_PATCH	DS threads are passed one patch, up to 8 domain point inputs, and up to 8 output vertex handles. All the domain points are associated with the single input patch. The DS kernel (at KSP) is expected to run in SIMD8 execution mode. The DUAL_PATCH KSP is ignored.	
	2h	SIMD8_SINGLE_OR_DUAL_PATCH	This mode enables use of both the KSP and the DUAL_PATCH KSP. The KSP kernel operates just like in SIMD8_SINGLE_PATCH mode. See DUAL_PATCH Thread Execution for a discussion of how the DUAL_PATCH KSP is used.	At least 2 HS URB handles must be allocated in order to enable this mode.
	3h	Reserved		
<b>Programming Notes</b>				
SIMD4X2 mode is no longer allowed.				

<b>3DSTATE_DS_BODY</b>						
2	<p><b>Compute W Coordinate Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, the DS unit will (for each domain point) compute <math>W = 1 - (U + V)</math> and pass the result as a floating point value in the DS thread payload. If DISABLED, 0.0 will be passed. This field must only be ENABLED for the tessellation of TRI domains, where UVW coordinates are required. This field must be DISABLED for other domains (as they only require UV coordinates) otherwise the computed W coordinate is UNDEFINED. This field is ignored if DS Function Enable is DISABLED.</p>	Format:	Enable			
	Format:	Enable				
	<p><b>Cache Disable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Disable</td> </tr> </table> <p>This bit controls the operation of the DS Cache. This field is ignored if DS Function Enable is DISABLED. If the DS Cache is DISABLED and the DS Function is ENABLED, the DS Cache is not used and all incoming domain points will be passed to DS threads. If the DS Cache is ENABLED and the DS Function is ENABLED, incoming domain points that do not hit in the DS Cache will be passed to DS threads. The DS Cache is invalidated whenever the DS Cache becomes DISABLED, whenever the DS Function Enable toggles, and between patches.</p>	Format:	Disable			
Format:	Disable					
<p><b>Function Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, DS threads will be spawned to process incoming domain points which miss in the DS cache. If DISABLED, the DS stage goes into pass-through mode and performs no specific processing. This field is always used.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </table> <p>The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.</p>	Format:	Enable	Programming Notes			
Format:	Enable					
Programming Notes						
7	<p><b>31:27 Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO				
	Format:	MBZ				
	<p><b>26:21 Vertex URB Entry Output Read Offset</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>U6</td> </tr> </table> <p>Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB by SBE.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <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;">[0,63]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,63]
Format:	U6					
Value	Name					
[0,63]						
<p><b>20:16 Vertex URB Entry Output Length</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>U5</td> </tr> </table> <p>Specifies the amount of URB data written for each Vertex URB entry, in 256-bit register increments.</p>	Format:	U5				
Format:	U5					

<b>3DSTATE_DS_BODY</b>										
		<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;">[1,16]</td> <td></td> </tr> <tr> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="2">This length does not include the vertex header.</td> </tr> </tbody> </table>	Value	Name	[1,16]		<b>Programming Notes</b>		This length does not include the vertex header.	
Value	Name									
[1,16]										
<b>Programming Notes</b>										
This length does not include the vertex header.										
	15:8	<b>User Clip Distance Clip Test Enable Bitmask</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%; text-align: center;">U8</td> </tr> </table> <p>This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept / must clip determination needs to be made. DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.</p>	Format:	U8						
	Format:	U8								
7:0	<b>User Clip Distance Cull Test Enable Bitmask</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%; text-align: center;">U8</td> </tr> </table> <p>This 8 bit mask field selects which of the 8 user clip distances against which trivial reject / trivial accept determination needs to be made (does not cause a must clip). DX10 allows simultaneous use of ClipDistance and Cull Distance test of up to 8 distances.</p>	Format:	U8							
Format:	U8									
8..9	63:6	<b>DUAL_PATCH Kernel Start Pointer</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td style="width: 70%;">InstructionBaseOffset[63:6]</td> </tr> </table> <p>This field specifies the starting location of the DUAL_PATCH kernel program run by threads spawned by this FF unit. It is specified as a 64-byte-granular offset from the Instruction Base Address. This field is ignored if DS Function Enable is DISABLED. See DUAL_PATCH Thread Execution for a discussion of how the DUAL_PATCH KSP is used.</p>	Format:	InstructionBaseOffset[63:6]						
Format:	InstructionBaseOffset[63:6]									
	5:0	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="width: 40%; text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO									
Format:	MBZ									

## 3DSTATE\_GS\_BODY

3DSTATE_GS_BODY													
Source:	RenderCS												
Size (in bits):	288												
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000												
DWord	Bit	Description											
0..1	63:6	<p><b>Kernel Start Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[63:6]</td> </tr> </table> <p>This field specifies the starting location of the kernel program run by threads spawned by this FF unit. It is specified as a 64-byte-granular offset from the Instruction Base Address.</p>	Format:	InstructionBaseOffset[63:6]									
	Format:	InstructionBaseOffset[63:6]											
5:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
Access:	RO												
Format:	MBZ												
2	31	<p><b>Single Program Flow</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Specifies the initial condition of the kernel program as either a single program flow (SIMDn<sub>xm</sub> with m = 1) or as multiple program flows (SIMDn<sub>xm</sub> with m &gt; 1). See CR0 description in ISA Execution Environment.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable</td> <td>Single Program Flow disabled</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Single Program Flow enabled</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0h	Disable	Single Program Flow disabled	1h	Enable	Single Program Flow enabled
	Format:	Enable											
Value	Name	Description											
0h	Disable	Single Program Flow disabled											
1h	Enable	Single Program Flow enabled											
30	<p><b>Vector Mask Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Upon subsequent GS thread dispatches, this bit is loaded into the EU's Vector Mask Enable (VME, cr0.0[3]) thread state. Refer to EU documentation for the definition and use of VME state.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Dmask</td> <td>The EU will use the Dispatch Mask (supplied by the GS stage) for instruction execution.</td> </tr> <tr> <td>1h</td> <td>Vmask</td> <td>The EU will use the Vector Mask (derived from Dispatch Mask) for instruction execution.</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>Under normal conditions SW shall specify DMask, as the GS stage will provide a Dispatch Mask appropriate to SIMD4x2 or SIMD8 thread execution (as a function of dispatch mode). E.g., for SIMD4x2 execution, the GS stage will generate a Dispatch Mask that is equal to what the EU would use as the Vector Mask. For SIMD8 execution there is no known usage model for use of Vector Mask (as there is for PS shaders).</p>	Format:	Enable	Value	Name	Description	0h	Dmask	The EU will use the Dispatch Mask (supplied by the GS stage) for instruction execution.	1h	Vmask	The EU will use the Vector Mask (derived from Dispatch Mask) for instruction execution.	
Format:	Enable												
Value	Name	Description											
0h	Dmask	The EU will use the Dispatch Mask (supplied by the GS stage) for instruction execution.											
1h	Vmask	The EU will use the Vector Mask (derived from Dispatch Mask) for instruction execution.											

## 3DSTATE\_GS\_BODY

29:27	<p><b>Sampler Count</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U3</td> </tr> </table> <p>Specifies how many samplers (in multiples of 4) the geometry shader kernel uses. Used only for prefetching the associated sampler state entries.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 55%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No Samplers</td> <td>No Samplers used</td> </tr> <tr> <td>1h</td> <td>1-4 Samplers</td> <td>Between 1 and 4 samplers used</td> </tr> <tr> <td>2h</td> <td>5-8 Samplers</td> <td>Between 5 and 8 samplers used</td> </tr> <tr> <td>3h</td> <td>9-12 Samplers</td> <td>Between 9 and 12 samplers used</td> </tr> <tr> <td>4h</td> <td>13-16 Samplers</td> <td>Between 13 and 16 samplers used</td> </tr> <tr> <td>5h-7h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U3	Value	Name	Description	0h	No Samplers	No Samplers used	1h	1-4 Samplers	Between 1 and 4 samplers used	2h	5-8 Samplers	Between 5 and 8 samplers used	3h	9-12 Samplers	Between 9 and 12 samplers used	4h	13-16 Samplers	Between 13 and 16 samplers used	5h-7h	Reserved	
Format:	U3																							
Value	Name	Description																						
0h	No Samplers	No Samplers used																						
1h	1-4 Samplers	Between 1 and 4 samplers used																						
2h	5-8 Samplers	Between 5 and 8 samplers used																						
3h	9-12 Samplers	Between 9 and 12 samplers used																						
4h	13-16 Samplers	Between 13 and 16 samplers used																						
5h-7h	Reserved																							
26	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																			
Access:	RO																							
Format:	MBZ																							
25:18	<p><b>Binding Table Entry Count</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>When <b>HW Generated Binding Table</b> is disabled: Specifies how many binding table entries the kernel uses. Used only for prefetching of the binding table entries and associated surface state. Note: For kernels using a large number of binding table entries, it may be wise to set this field to zero to avoid prefetching too many entries and thrashing the state cache. When <b>HW Generated Binding Table</b> bit is enabled: This field indicates which cache lines (512bit units - 32 Binding Table Entry section) should be fetched. Each bit in this field corresponds to a cache line. Only the 1st 4 non-zero Binding Table entries of each 32 Binding Table entry section prefetched will have its surface state prefetched.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 100%;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.</td> </tr> </tbody> </table>	Format:	U8	Programming Notes	When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.																			
Format:	U8																							
Programming Notes																								
When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.																								
17	<p><b>Thread Dispatch Priority</b></p> <p>Specifies the priority of the thread for dispatch.</p> <table border="1" style="width: 100%; text-align: center;"> <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>Normal</td> <td>Normal thread dispatch priority</td> </tr> <tr> <td>1h</td> <td>High</td> <td>High thread dispatch priority</td> </tr> </tbody> </table>	Value	Name	Description	0h	Normal	Normal thread dispatch priority	1h	High	High thread dispatch priority														
Value	Name	Description																						
0h	Normal	Normal thread dispatch priority																						
1h	High	High thread dispatch priority																						
16	<p><b>Floating Point Mode</b></p> <p>Specifies the initial floating point mode used by the dispatched thread.</p> <table border="1" style="width: 100%; text-align: center;"> <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>IEEE-754</td> <td>Use IEEE-754 Rules</td> </tr> <tr> <td>1h</td> <td>Alternate</td> <td>Use alternate rules</td> </tr> </tbody> </table>	Value	Name	Description	0h	IEEE-754	Use IEEE-754 Rules	1h	Alternate	Use alternate rules														
Value	Name	Description																						
0h	IEEE-754	Use IEEE-754 Rules																						
1h	Alternate	Use alternate rules																						

<b>3DSTATE_GS_BODY</b>								
	15:14	<b>Reserved</b>	Access: RO	Format: MBZ				
	13	<b>Illegal Opcode Exception Enable</b>	Format: Enable	This bit gets loaded into EU CR0.1[12] (note the bit # difference). See <i>Exceptions and ISA Execution Environment</i> .				
	12	<b>Accesses UAV</b>	Format: Enable	This field must be set when GS has a UAV access. <b>Programming Notes</b> This field must not be set when GS Function Enable is disabled.				
	11	<b>Mask Stack Exception Enable</b>	Format: Enable	This bit gets loaded into EU CR0.1[11]. See <i>Exceptions and ISA Execution Environment</i> .				
	10:8	<b>Reserved</b>	Format: MBZ					
	7	<b>Software Exception Enable</b>	Format: Enable	This bit gets loaded into EU CR0.1[13] (note the bit # difference). See <i>Exceptions and ISA Execution Environment</i> .				
	6	<b>Reserved</b>	Access: RO	Format: MBZ				
	5:0	<b>Expected Vertex Count</b>	Format: U6	Specifies the number of vertices per input object expected by the GS thread. Input topologies not matching this expect value are discarded. Note that <b>DiscardAdjacency</b> is also considered (e.g., if the value programmed is 3 and DiscardAdjacency is set, TRILIST_ADJ and TRISTRIP_ADJ topologies are <u>not</u> discarded as they will pass 3 vertices/object to the GS threads).				
			<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>[1,32]</td><td></td></tr></tbody></table>	Value	Name	[1,32]		
	Value	Name						
[1,32]								
3.4	63:32	<b>Reserved</b>	Access: RO	Format: MBZ				

<b>3DSTATE_GS_BODY</b>								
	<b>31:10 Scratch Space Base Pointer</b> <table border="1"> <tr> <td>Format:</td> <td>GeneralStateOffset[31:10]</td> </tr> </table> <p>Specifies the starting location of the scratch space area allocated to this FF unit as a 1K-byte aligned offset from the General State Base Address. If required, each thread spawned by this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space. The computed offset of the thread-specific portion will be passed in the thread payload as Scratch Space Offset. The thread is expected to utilize "stateless" DataPort read/write requests to access scratch space, where the DataPort will cause the General State Base Address to be added to the offset passed in the request header. This field is ignored if VS Function Enable is DISABLED.</p>	Format:	GeneralStateOffset[31:10]					
	Format:	GeneralStateOffset[31:10]						
	<b>9:4 Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO							
Format:	MBZ							
<b>3:0 Per-Thread Scratch Space</b> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Specifies the amount of scratch space to be allocated to each thread spawned by this FF unit. The driver must allocate enough contiguous scratch space, starting at the Scratch Space Base Pointer, to ensure that the Maximum Number of Threads can each get Per-Thread Scratch Space size without exceeding the driver-allocated scratch space.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td></td> <td>indicating [1K Bytes, 2M Bytes]</td> </tr> </tbody> </table>	Format:	U4	Value	Name	Description	[0,11]		indicating [1K Bytes, 2M Bytes]
Format:	U4							
Value	Name	Description						
[0,11]		indicating [1K Bytes, 2M Bytes]						
5	<b>31 Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
	Access:	RO						
	Format:	MBZ						
<b>30:29 Dispatch GRF Start Register For URB Data [5:4]</b> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Specifies bit [5:4] of the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. The <b>Dispatch GRF Start Register For URB Data [3:0]</b> field is used to specify bits [3:0] of the starting GRF register number.</p>	Format:	U2						
Format:	U2							
<b>28:23 Output Vertex Size</b> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>[0,63] indicating [1,64] 16B units</p> <p>Specifies the size of each vertex stored in the GS output entry (following any Control Header data) as a number of 128-bit units (minus one).</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>Programming Restrictions: The vertex size must be programmed as a multiple of 32B units with the following exception: Rendering is disabled (as per SOL stage state) and the vertex size output by the GS thread is 16B. If rendering is enabled (as per SOL state) the vertex size must be</td> </tr> </tbody> </table>	Format:	U6	Programming Notes	Programming Restrictions: The vertex size must be programmed as a multiple of 32B units with the following exception: Rendering is disabled (as per SOL stage state) and the vertex size output by the GS thread is 16B. If rendering is enabled (as per SOL state) the vertex size must be				
Format:	U6							
Programming Notes								
Programming Restrictions: The vertex size must be programmed as a multiple of 32B units with the following exception: Rendering is disabled (as per SOL stage state) and the vertex size output by the GS thread is 16B. If rendering is enabled (as per SOL state) the vertex size must be								

<b>3DSTATE_GS_BODY</b>														
		programmed as a multiple of 32B units. In other words, the only time software can program a vertex size with an odd number of 16B units is when rendering is disabled.												
22:17	<b>Output Topology</b>	<table border="1"> <tr> <td>Format:</td> <td><b>3D_Prim_Topo_Type</b></td> </tr> </table> <p>This field specifies the topology type (3DPrimType) to be associated with GS-thread output vertices (if any).</p>	Format:	<b>3D_Prim_Topo_Type</b>										
Format:	<b>3D_Prim_Topo_Type</b>													
16:11	<b>Vertex URB Entry Read Length</b>	<p>Specifies the amount of URB data read and passed in the thread payload for each Vertex URB entry, in 256-bit register increments.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Programming Restriction: This field must be a non-zero value if Include Vertex Handles is cleared to zero.</td> </tr> </table>	Programming Notes		Programming Restriction: This field must be a non-zero value if Include Vertex Handles is cleared to zero.									
Programming Notes														
Programming Restriction: This field must be a non-zero value if Include Vertex Handles is cleared to zero.														
10	<b>Include Vertex Handles</b>	<table border="1"> <tr> <td>Format:</td> <td>Boolean</td> </tr> </table> <p>If set, all the input Vertex URB handles are included in the payload. These are referred to as "pull model" URB handles, as the thread will use them to read from the URB.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Programming Restriction: This field must be set if Vertex URB Entry Read Length is cleared to zero.</td> </tr> </table>	Format:	Boolean	Programming Notes		Programming Restriction: This field must be set if Vertex URB Entry Read Length is cleared to zero.							
Format:	Boolean													
Programming Notes														
Programming Restriction: This field must be set if Vertex URB Entry Read Length is cleared to zero.														
9:4	<b>Vertex URB Entry Read Offset</b>	<table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all Vertex URB entries passed to the thread.</p>	Format:	U6										
Format:	U6													
3:0	<b>Dispatch GRF Start Register For URB Data</b>	<table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload.</p> <p>The Dispatch GRF Start Register for URB Data [5:4] field is used to extend the range of the starting GRF register number to [0,63].</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,15]</td> <td></td> <td>indicating bits [3:0] of the GRF number</td> </tr> </tbody> </table> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If Include Vertex Handles is enabled (pull or hybrid handles case), then For simd4x2: For DUAL_OBJECT dispatch mode this field should be:</td> </tr> </table>	Format:	U4	Value	Name	Description	[0,15]		indicating bits [3:0] of the GRF number	Programming Notes		If Include Vertex Handles is enabled (pull or hybrid handles case), then For simd4x2: For DUAL_OBJECT dispatch mode this field should be:	
Format:	U4													
Value	Name	Description												
[0,15]		indicating bits [3:0] of the GRF number												
Programming Notes														
If Include Vertex Handles is enabled (pull or hybrid handles case), then For simd4x2: For DUAL_OBJECT dispatch mode this field should be:														



<b>3DSTATE_GS_BODY</b>			
		$(((2 * \text{numVerticesPerObject}) + 8 - 1) / 8) + 1$ For SINGLE and DUAL_INSTANCE dispatch modes this field should be: $((\text{numVerticesPerObject} + 8 - 1) / 8) + 1$ If Include Primitive ID is set, then add 1 to the value obtained by using the above  If Include Vertex Handles is enabled (pull or hybrid handles case), then. simd8:For InstanceCount == 1:numVerticesPerObject 2For InstanceCount > 1:((numVerticesPerObject 8 - 1)/8) 2If Include Primitive ID is set, then add 1 to the value obtained by using the above	
6	31:24	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	23:20	<b>Control Data Header Size</b>	
		Format:	U4
		Specifies the number of 32B units of control data header located at the start of the GS URB entry. The value 0 indicates there is no control data header, and Control Data Format is ignored and neither Cut nor StreamID bits are defined. Software must ensure that the Control Data Header Size is sufficient to accommodate the maximum number of vertices possibly output by the GS thread. It is UNDEFINED for a GS thread to report more output vertices than can be accommodated in a non-zero-sized header.	
		<b>Value</b>	<b>Name</b>
		[0,8]	32B Units
19:15	<b>Instance Control</b>		
	Format:	U5-1	
	Specifies the number of instances (minus one) for each input object. To avoid confusion, this document uses the term " <b>InstanceCount</b> " to refer to InstanceControl+1, with a range of [1,32]If <b>InstanceCount</b> > 1, DUAL_OBJECT mode is invalid. Software will likely want to use DUAL_INSTANCE mode for higher performance, but SINGLE mode is also supported. When <b>InstanceCount</b> =1 (one instance per object), software can decide which dispatch mode to use. DUAL_OBJECT mode would likely be the best choice for performance, followed by SINGLE mode. DUAL_INSTANCE mode is not recommended but is supported.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	[0,31]		Indicating [1,31] instances
14:13	<b>Default Stream Id</b>		
	Format:	U2	
	When the GS is enabled, unless the GS output entry contains StreamID bits in the control header, this field specifies the default StreamID associated with any GS-thread output vertices. When the GS is disabled, StreamID will be output as 0.		
12:11	<b>Dispatch Mode</b>		
	Format:	U2	
	This field specifies how the GS unit dispatches multiple instances and/or multiple objects.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	3h	SIMD8	Each thread shades up to 8
			<b>Programming Notes</b>
			The driver must send pipe control with a cs

## 3DSTATE\_GS\_BODY

		different objects or (if InstanceCount > 1) 8 instances of a single object.	stall after a 3dstate_gs state change and the Dispatch Mode is simd8 and the number of handles allocated to gs is less than 16.
<b>Programming Notes</b>			
The GS must be allocated at least two URB handles or behavior is UNDEFINED for Dual Instance or Dual Object mode.			
The only valid Dispatch Mode is SIMD8.			
10	<b>Statistics Enable</b>		
Format:		Enable	
This bit controls whether GS-unit-specific statistics register(s) can be incremented.			
<b>Value</b>	<b>Name</b>	<b>Description</b>	
0h	Disable	GS_INVOCATIONS_COUNT and GS_PRIMITIVES_COUNT cannot increment	
1h	Enable	GS_INVOCATIONS_COUNT and GS_PRIMITIVES_COUNT can increment	
9:5	<b>Invocations Increment Value</b>		
Format:		U5	
Specifies how much to increment the GS_INVOCATIONS_COUNT for each instance of each object. This control is provided to allow software to process multiple instances (from an API POV) in a single kernel invocation. In SINGLE dispatch mode, the counter will increment by this value for each dispatch (as it's only one instance of one object). In DUAL_INSTANCE mode, the counter will be incremented by the value if only one instance is included in the dispatch (i.e., the last odd instance), otherwise the counter will be incremented by twice this value. In DUAL_OBJECT dispatch mode, the counter will be incremented by the value if only one object is included in the dispatch (i.e., a forced dispatch of one object), otherwise the counter will be incremented by twice this value.			
<b>Value</b>	<b>Name</b>	<b>Description</b>	
[0,31]		indicating an increment of [1,32]	
4	<b>Include Primitive ID</b>		
Format:		Boolean	
If set, R1 of the payload is written with Primitive ID value(s). If clear, these Primitive ID values are not included in the payload R1.			
3	<b>Hint</b>		
Format:		U1	
This state bit is simply passed in GS thread payloads for use by the GS kernel - it has no other impact on hardware operation.			
2	<b>Reorder Mode</b>		
This bit controls how vertices of triangle objects resulting from TRISTRIP[_ADJ][_REV] topologies are [re]ordered when passed in the GS thread payload See Object Vertex Ordering table (below).			
<b>Value</b>	<b>Name</b>	<b>Description</b>	
0h	LEADING	Reorder the vertices of alternating triangles of a TRISTRIP[_ADJ] such that the leading (first) vertices are in consecutive order starting at v0. A similar	

<b>3DSTATE_GS_BODY</b>													
		reordering is performed on alternating triangles in a TRISTRIP_REV.											
	1h	TRAILING Reorder the vertices of alternating triangles of a TRISTRIP[_ADJ] such that the trailing (last) vertices are in consecutive order starting at v2. A similar reordering is performed on alternating triangles in a TRISTRIP_REV.											
	1	<p><b>Discard Adjacency</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>When set, adjacent vertices <u>will not be passed</u> in the GS payload when objects with adjacency are processed. Instead, only the non-adjacent vertices will be passed in the same fashion as the without-adjacency form of the primitive. Software should set this bit whenever a GS kernel is used that <u>does not expect</u> adjacent vertices. This allows both with-adjacency/without-adjacency variants of the primitive to be submitted to the pipeline (via 3DPRIMITIVE) - the GS unit will silently discard any adjacent vertices and present the GS thread with only the internal object. When clear, adjacent vertices <u>will be passed</u> to the GS thread, as dictated by the incoming primitive type. Software should only clear this bit when a GS kernel is used that does expect adjacent vertices. E.g., if the GS kernel is compiled to expect a TRIANGLE_ADJ object, software must clear this bit. Software should also clear this bit if the GS kernel expects a POINT or PATCHLIST_n object (which don't have with-adjacency variants).</p> <p>The only hardware assistance is to allow the submission of a with-adjacency variant of a primitive when operating with a GS kernel that expects the without-adjacency variant of the object. (E.g., when the GS kernel is compiled to expect a TRIANGLE object, software should set this bit just in case a TRILIST_ADJ is submitted to the pipeline.) Note that the GS unit is otherwise not aware of the object type that is expected by the GS kernel. It is up to software to ensure that the submitted primitive type (in 3DPRIMITIVE) is otherwise compatible with the object type expected by the GS kernel. (E.g., if the GS kernel expects a LINE_ADJ object, only LINELIST_ADJ or LINESTRIP_ADJ should be submitted, otherwise the GS kernel will produce unpredictable results.) Also note that it is possible to craft a GS kernel which can accept any object type that's thrown at it by first examining the PrimType passed in the payload and then using this info to correctly interpret the number of vertices passed in the payload.</p>	Format:	Enable									
Format:	Enable												
	0	<p><b>Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>Specifies whether the GS stage is enabled or disabled (pass-through).</p>	Format:	Enable									
Format:	Enable												
7	31	<p><b>Control Data Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U1</td> </tr> </table> <p>This field specifies the format of the control data header (if any).</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 75%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>CUT</td> <td>The control data header contains Cut bits.</td> </tr> <tr> <td>1h</td> <td>SID</td> <td>The control data header contains StreamID bits. Output Topology must be set to POINTLIST, or behavior is UNDEFINED.</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	0h	CUT	The control data header contains Cut bits.	1h	SID	The control data header contains StreamID bits. Output Topology must be set to POINTLIST, or behavior is UNDEFINED.
Format:	U1												
Value	Name	Description											
0h	CUT	The control data header contains Cut bits.											
1h	SID	The control data header contains StreamID bits. Output Topology must be set to POINTLIST, or behavior is UNDEFINED.											

## 3DSTATE\_GS\_BODY

	30	<b>Static Output</b>	Format:	Enable						
	<p>Specifies whether the GS shader outputs a static number of vertices per invocation. If this bit is clear, the number of vertices output by each GS shader invocation is stored by the GS thread in Vertex Count at the very beginning of the output URB entry (see GS URB Entry description).</p>									
	29:27	<b>Reserved</b>	Access:	RO						
	<p>Format: MBZ</p>									
	26:16	<b>Static Output Vertex Count</b>	Format:	U11						
	<p>If <b>GSEnable</b> is ENABLED and <b>StaticOutput</b> is ENABLED, this field specifies the total number of vertices output each GS shader invocation. If <b>GSEnable</b> is ENABLED and <b>StaticOutput</b> is DISABLED (i.e., variable GS output), the total number of vertices output by a GS shader invocation is stored by the thread at the very beginning of the output URB entry, and this field is ignored. (See GS URB Entry description ).</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;">[0,1024]</td> <td></td> </tr> </tbody> </table>					Value	Name	[0,1024]		
Value	Name									
[0,1024]										
	15:9	<b>Reserved</b>	Access:	RO						
	<p>Format: MBZ</p>									
	8:0	<b>Maximum Number of Threads</b>	Format:	U9-1						
	<p>Specifies the maximum number of simultaneous threads allowed to be active. Used to avoid using up the scratch space. Programming the value of the max threads over the number of threads based off number of threads supported in the execution units may improve performance since the architecture allows threads to be buffered between the check for max threads and the actual dispatch into the EU. Programming the max values to a number less than the number of threads supported in the execution units may reduce performance.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%; text-align: center;">Value</th> <th style="width: 20%; text-align: center;">Name</th> <th style="width: 60%; text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">[0,335]</td> <td></td> <td>indicating thread count of [1,336]</td> </tr> </tbody> </table>					Value	Name	Description	[0,335]	
Value	Name	Description								
[0,335]		indicating thread count of [1,336]								
8	31:27	<b>Reserved</b>	Access:	RO						
	<p>Format: MBZ</p>									
	26:21	<b>Vertex URB Entry Output Read Offset</b>	Format:	U6						
	<p>Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB by SBE.</p> <table border="1" style="width: 100%; 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;">[0,63]</td> <td></td> </tr> </tbody> </table>					Value	Name	[0,63]		
Value	Name									
[0,63]										

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

## 3DSTATE\_HIER\_DEPTH\_BUFFER\_BODY

3DSTATE_HIER_DEPTH_BUFFER_BODY																													
Source:	RenderCS																												
Size (in bits):	128																												
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000																												
DWord	Bit	Description																											
0	31:25	<b>Hierarchical Depth Buffer Object Control State</b> Format: <b>MEMORY_OBJECT_CONTROL_STATE</b> Specifies the memory object control state for the hierarchical depth buffer.																											
	24	<b>Reserved</b> Access: RO Format: MBZ																											
	23:22	<b>Tiled Mode</b> <table border="1"> <thead> <tr> <th colspan="3">Description</th> </tr> </thead> <tbody> <tr> <td colspan="3">This field specifies the tiled resource mode.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>0h</td> <td>No tiled resource (Tile Y Mode)</td> <td></td> </tr> <tr> <td>1h</td> <td>TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="3">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="3">HZ buffer only supports Tile Y mode.</td> </tr> </tbody> </table>	Description			This field specifies the tiled resource mode.			Value	Name	Description	0h	No tiled resource (Tile Y Mode)		1h	TILEYF	4KB tiled resources	2h	TILEYS	64KB tiled resources	3h	Reserved		Programming Notes			HZ buffer only supports Tile Y mode.		
	Description																												
	This field specifies the tiled resource mode.																												
Value	Name	Description																											
0h	No tiled resource (Tile Y Mode)																												
1h	TILEYF	4KB tiled resources																											
2h	TILEYS	64KB tiled resources																											
3h	Reserved																												
Programming Notes																													
HZ buffer only supports Tile Y mode.																													
21	<b>Reserved</b> Access: RO Format: MBZ																												
20	<b>Write thru enable for Texture</b> <table border="1"> <thead> <tr> <th colspan="2">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">This bit must be set if the Depth buffer is used as a texture. If this bit is set, HZ will force a write of non-clear values to the Depth buffer avoiding the need of a Depth resolve. This means that HZ will not write planes to the HZ\$.</td> </tr> <tr> <th>Value</th> <th>Name</th> </tr> <tr> <td>1h</td> <td>Enable</td> </tr> <tr> <td>0h</td> <td>Disable <b>[Default]</b></td> </tr> </tbody> </table>	Description		This bit must be set if the Depth buffer is used as a texture. If this bit is set, HZ will force a write of non-clear values to the Depth buffer avoiding the need of a Depth resolve. This means that HZ will not write planes to the HZ\$.		Value	Name	1h	Enable	0h	Disable <b>[Default]</b>																		
Description																													
This bit must be set if the Depth buffer is used as a texture. If this bit is set, HZ will force a write of non-clear values to the Depth buffer avoiding the need of a Depth resolve. This means that HZ will not write planes to the HZ\$.																													
Value	Name																												
1h	Enable																												
0h	Disable <b>[Default]</b>																												

## 3DSTATE\_HIER\_DEPTH\_BUFFER\_BODY

		Programming Notes	
		<p>For MSAA with multiple resolves, SW may set this bit after the first resolve with compression disabled to enforce HW to write-through to depth buffer as uncompressed data to save on future resolves.</p> <p>However the clears on a surface must be done with this bit reset for MSAA.</p>	
	19:18	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	17	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	16:0	<b>Surface Pitch</b>	
		Format:	U17-1
		This field specifies the pitch of the hierarchical depth buffer in (#Bytes - 1).	
		<b>Value</b>	<b>Name</b>
		[7Fh,1FFFFh]	corresponding to [128B, 512KB] also restricted to a multiple of 128B
		<b>Programming Notes</b>	
		Since this surface is tiled, the pitch specified must be a multiple of the tile pitch, in the range [128B, 128KB].	
1..2	63:0	<b>Surface Base Address</b>	
		Format:	GraphicsAddress[63:0]
		This field specifies the address of the buffer in Graphics Memory.	
		<b>Programming Notes</b>	
		The Hierarchical Depth Buffer can only be mapped to Main Memory (uncached). If the surface is tiled, the base address must conform to the Per-Surface Tiling Alignment Rules as documented in TBD.	
3	31:16	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	15	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	14:0	<b>Surface QPitch</b>	
		Format:	U17[16:2]
		The interpretation of this field is dependent on <b>Surface Type</b> as follows:	
		<ul style="list-style-type: none"> <li>SURFTYPE_2D/CUBE: distance in <i>rows</i> between array slices</li> </ul>	

<b>3DSTATE_HIER_DEPTH_BUFFER_BODY</b>		
Format: QPitch[16:2]		
Value	Name	Description
[1h,7FFFh]		in multiples of 4 (low 2 bits missing)
<b>Programming Notes</b>		
This field must be set to an integer multiple of 16 (QPitch[3] MBZ) Software must ensure that this field is set to a value sufficiently large such that the array slices in the surface do not overlap. Refer to the Memory Data Formats section for information on how surfaces are stored in memory.		



## 3DSTATE\_HS\_BODY

3DSTATE_HS_BODY				
Source:	RenderCS			
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:30	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	29:27	<b>Sampler Count</b>		
		Format:	U3	
		Specifies how many samplers (in multiples of 4) the HS kernels use. Used only for prefetching the associated sampler state entries.		
		Value	Name	Description
		0h	No Samplers	no samplers used
		1h	1-4 Samplers	between 1 and 4 samplers used
		2h	5-8 Samplers	between 5 and 8 samplers used
3h		9-12 Samplers	between 9 and 12 samplers used	
4h	13-16 Samplers	between 13 and 16 samplers used		
5h-7h	Reserved	Reserved		
26	<b>Reserved</b>			
	Access:	RO		
	Format:	MBZ		
25:18	<b>Binding Table Entry Count</b>			
	Format:	U8		
	When HW Generated Binding Table is disabled: Specifies how many binding table entries the kernel uses. Used only for prefetching of the binding table entries and associated surface state. Note: For kernels using a large number of binding table entries, it may be wise to set this field to zero to avoid prefetching too many entries and thrashing the state cache.			
	Programming Notes			
	When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.			
17	<b>Thread Dispatch Priority</b>			
	Specifies the priority of the thread for dispatch			
	Value	Name	Description	
	0h	Normal	Normal Priority	
1h	High	High Priority		

## 3DSTATE\_HS\_BODY

	16	<b>Floating Point Mode</b> Specifies the initial floating point mode used by the dispatched thread. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>IEEE-754</td> <td>Use IEEE-754 Rules</td> </tr> <tr> <td>1h</td> <td>alternate</td> <td>Use alternate rules</td> </tr> </tbody> </table>	Value	Name	Description	0h	IEEE-754	Use IEEE-754 Rules	1h	alternate	Use alternate rules
	Value	Name	Description								
	0h	IEEE-754	Use IEEE-754 Rules								
	1h	alternate	Use alternate rules								
	15:14	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
	Access:	RO									
	Format:	MBZ									
	13	<b>Illegal Opcode Exception Enable</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.</p>	Format:	Enable							
	Format:	Enable									
	12	<b>Software Exception Enable</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit gets loaded into EU CRO1[13] (note the bit # difference). See Exceptions and ISA Execution Environment.</p>	Format:	Enable							
Format:	Enable										
11	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
Access:	RO										
Format:	MBZ										
10:8	<b>Reserved</b> <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										
7:4	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
Access:	RO										
Format:	MBZ										
3:0	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
Access:	RO										
Format:	MBZ										
1	31 <b>Enable</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Specifies whether the HS function is enabled or disabled (pass-through). If ENABLED MI_TOPOLOGY_FILTER must be used to silently discard any topologies that the HS kernel is not expecting. E.g., if the HS kernel is expecting PATCHLIST_32 topologies, MI_TOPOLOGY_FILTER must be set to PATCHLIST_32 so only those topologies can reach the enabled HS.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.</td> </tr> </tbody> </table>	Format:	Enable	Programming Notes	The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.						
Format:	Enable										
Programming Notes											
The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.											

<b>3DSTATE_HS_BODY</b>									
	30	<b>Reserved</b>							
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO								
Format:	MBZ								
	29	<b>Statistics Enable</b>							
		<table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit controls whether HS-unit-specific statistics register(s) will increment (for each patch).</p>	Format:	Enable					
Format:	Enable								
	28:17	<b>Reserved</b>							
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO								
Format:	MBZ								
16:8		<b>Maximum Number of Threads</b>							
		<table border="1"> <tr> <td>Format:</td> <td>U9-1</td> </tr> </table> <p>Specifies the maximum number of simultaneous threads allowed to be active. Used to avoid using up the scratch space. Programming the value of the max threads over the number of threads based off number of threads supported in the execution units may improve performance since the architecture allows threads to be buffered between the check for max threads and the actual dispatch into the EU. Programming the max values to a number less than the number of threads supported in the execution units may reduce performance.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,335]</td> <td></td> <td>indicating thread count of [1,336]</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>The Maximum Number of Threads must be set to at least twice the setting of 3DSTATE_HS::<b>Instance Count</b>.</p>	Format:	U9-1	Value	Name	Description	[0,335]	
Format:	U9-1								
Value	Name	Description							
[0,335]		indicating thread count of [1,336]							
	7:5	<b>Reserved</b>							
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO								
Format:	MBZ								
2..3	63:6	<b>Instance Count</b>							
		<table border="1"> <tr> <td>Format:</td> <td>U5-1</td> </tr> </table> <p>This field determines the number of threads (minus one) spawned per input patch. If the HS kernel uses a barrier function, software must restrict the <b>Instance Count</b> to the number of threads that can be simultaneously active within a subslice. Factors which must be considered includes scratch memory availability.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td></td> <td>representing [1,32] instances</td> </tr> </tbody> </table>	Format:	U5-1	Value	Name	Description	[0,31]	
Format:	U5-1								
Value	Name	Description							
[0,31]		representing [1,32] instances							
	63:6	<b>Kernel Start Pointer</b>							
		<table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[63:6]</td> </tr> </table> <p>This field specifies the starting location of the kernel program run by threads spawned by this FF unit. It is specified as a 64-byte-granular offset from the Instruction Base Address.</p>	Format:	InstructionBaseOffset[63:6]					
Format:	InstructionBaseOffset[63:6]								

<b>3DSTATE_HS_BODY</b>							
	5:0	<b>Reserved</b>					
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO						
Format:	MBZ						
4..5	63:32	<b>Reserved</b>					
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO					
	Format:	MBZ					
	31:10	<b>Scratch Space Base Pointer</b>					
Format: GeneralStateOffset[31:10]							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td></td> <td>Specifies the location of the scratch space area allocated to this FF unit, specified as a 1KB-granular offset from the General State Base Address. If required, each thread spawned by this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space.</td> </tr> </tbody> </table>		Value	Name	Description	[0,31]		Specifies the location of the scratch space area allocated to this FF unit, specified as a 1KB-granular offset from the General State Base Address. If required, each thread spawned by this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space.
Value	Name	Description					
[0,31]		Specifies the location of the scratch space area allocated to this FF unit, specified as a 1KB-granular offset from the General State Base Address. If required, each thread spawned by this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space.					
9:4	<b>Reserved</b>						
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
Access:	RO						
Format:	MBZ						
3:0	<b>Per-Thread Scratch Space</b>						
	Format: U4						
	<p>Specifies the amount of scratch space to be allocated to each thread spawned by this FF unit. The driver must allocate enough contiguous scratch space, starting at the Scratch Space Base Pointer, to ensure that the Maximum Number of Threads can each get Per-Thread Scratch Space size without exceeding the driver-allocated scratch space.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td></td> <td>Indicating[1K Bytes, 2M Bytes</td> </tr> </tbody> </table>		Value	Name	Description	[0,11]	
Value	Name	Description					
[0,11]		Indicating[1K Bytes, 2M Bytes					
6	31:29	<b>Reserved</b>					
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO					
	Format:	MBZ					
	28	<b>Dispatch GRF Start Register For URB Data [5]</b>					
		Format: U1					
		<p>Specifies bit [5] of the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. The <b>Dispatch GRF Start Register For URB Data [4:0]</b> field is used to specify bits [4:0] of the starting GRF register number.</p>					
27	<b>Single Program Flow</b>						
	Format: Enable						
		<p>Specifies the initial condition of the kernel program as either a single program flow (SIMDn<sub>xm</sub> with m = 1) or as multiple program flows (SIMDn<sub>xm</sub> with m &gt; 1). See CR0 description in <i>ISA Execution Environment</i>.</p>					

## 3DSTATE\_HS\_BODY

		Value	Name	Description											
		0h	Reserved												
		1h	Enable	Single Program Flow Enabled											
26	<b>Vector Mask Enable</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Upon subsequent HS thread dispatches, this bit is loaded into the EU's Vector Mask Enable (VME, cr0.0[3]) thread state. Refer to the EU documentation for the definition and use of VME state.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Dmask</td> <td>The EU will use the Dispatch Mask (supplied by the HS stage) for instruction execution.</td> </tr> <tr> <td>1h</td> <td>Vmask</td> <td>The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>Under normal conditions SW shall specify DMask, as the HS stage will provide a Dispatch Mask appropriate to SIMD4x2 or SIMD8 thread execution (as a function of dispatch mode). E.g., for SIMD4x2 thread execution, the HS state will generate a Dispatch Mask that is equal to what the EU would use as a Vector Mask. For SIMD8 execution there is no known usage model for use of Vector Mask (as there is for PS shaders).</p>			Format:	Enable	Value	Name	Description	0h	Dmask	The EU will use the Dispatch Mask (supplied by the HS stage) for instruction execution.	1h	Vmask	The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.
Format:	Enable														
Value	Name	Description													
0h	Dmask	The EU will use the Dispatch Mask (supplied by the HS stage) for instruction execution.													
1h	Vmask	The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.													
25	<b>Accesses UAV</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field must be set when HS has a UAV access</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This field must not be set when HS Function Enable is disabled.</p>			Format:	Enable									
Format:	Enable														
24	<b>Include Vertex Handles</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Boolean</td> </tr> </table> <p>If set, all the input Vertex URB handles are included in payloads. This field is ignored if <b>HS Function Enable</b> is DISABLED.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p><b>Programming Restriction:</b> This field must be set if value if <b>Vertex URB Entry Read Length</b> is cleared to zero.</p>			Format:	Boolean									
Format:	Boolean														
23:19	<b>Dispatch GRF Start Register For URB Data</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U5</td> </tr> </table> <p>Specifies the starting GRF register number for the URB portion (Constant + Vertices) of the thread payload. This field is ignored if <b>HS Function Enable</b> is DISABLED.</p> <p>The Dispatch GRF Start Register for URB Data [5] field is used to extend the range of the starting GRF register number to [0,63].</p>			Format:	U5									
Format:	U5														

## 3DSTATE\_HS\_BODY

		Value	Name	Description
		[0,31]		indicating bits [4:0] of the GRF number
		<b>Programming Notes</b>		
		When Include Vertex Handles is set for non-instanced 8_PATCH dispatch of PATCHLIST_30..32 objects, pushed vertex data and/or pushed constants cannot be used as they would need to start in the payload beyond the range of this field (i.e., beyond R31). When Include PrimitiveID is also set, this issue extends to non-instanced 8_PATCH dispatch of PATCHLIST_29..32 objects.		
18:17	<b>Dispatch Mode</b>	Format: <span style="float: right;">U2</span>		
		This field is unused to set the current thread dispatch mode for the HS stage.		
		Value	Name	Description
		2h	8_PATCH	HS threads are passed inputs and an output handle associated with (up to) 8 patches in SIMD8 fashion. See 8_PATCH Thread Payload.
		3h	Reserved	
		<b>Programming Notes</b>		
		DUAL_PATCH is not supported.		
16:11	<b>Vertex URB Entry Read Length</b>	Format: <span style="float: right;">U6</span>		
		Specifies the amount of URB data read and passed in the thread payload <u>for each Vertex URB entry</u> , in 256-bit register increments. This field is ignored if HS Function Enable is DISABLED.		
		Value	Name	Description
		[0,63]		
		<b>Programming Notes</b>		
		<b>Programming Restriction:</b> This field must be a non-zero value if <b>Include Vertex Handles</b> is cleared to zero.		
10	<b>Reserved</b>	Access: <span style="float: right;">RO</span>		
		Format: <span style="float: right;">MBZ</span>		
9:4	<b>Vertex URB Entry Read Offset</b>	Format: <span style="float: right;">U6</span>		
		Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all Vertex URB entries passed to the thread. This field is ignored if HS Function Enable is DISABLED.		
		Value	Name	Description
		[0,63]		

<b>3DSTATE_HS_BODY</b>		
3:1	<b>Patch Count Threshold</b>	
	Format:	U3
<p>Specifies the maximum number of patches that will be accumulated before a thread is dispatched. The dispatch of threads can (optionally) be forced before a full complement of eight patches have been accumulated.</p>		
<b>Value      Name      Description</b>		
[1,7]		
This specifies the maximum number of patches that will be accumulated before a thread dispatch is forced.		
0		
<b>[Default]</b> Early thread dispatch due to the Patch Count Threshold is disabled. A full complement of 8 patches can be accumulated before a thread is dispatched.		
0	<b>Include Primitive ID</b>	
	Format:	Enable
<p>If set, R1 of the payload is written with Primitive ID value(s). If clear, these Primitive IDs are not included in the payload R1.</p>		
<b>Programming Notes</b>		
<p>This field is only used when DUAL_PATCH DispatchMode is specified. In SINGLE_PATCH, the single Primitive ID is always passed in R0.</p>		
7	<b>31:0 Reserved</b>	
	Access:	RO
Format:		MBZ

## 3DSTATE\_INDEX\_BUFFER\_BODY

3DSTATE_INDEX_BUFFER_BODY			
Source:	RenderCS		
Size (in bits):	128		
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000		
DWord	Bit	Description	
0	31:12	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	11	<b>L3 Bypass Disable</b>	
		Format:	Disable
		<p style="text-align: center;"><b>Description</b></p> <p>When set, index data fetches are routed through the L3 and therefore that index data may be coherent with the L3 cache, as controlled by the Memory Object Control State (MOCS) value. I.e., if portions of the index buffer already reside in the L3 (e.g., were written or read by another L3 agent), reads from VF may hit in the L3 with the cached data returned to VF. If reads from VF miss in the L3 cache, the reads are directed to the next higher in the memory hierarchy, but the data returned is not placed in the L3 cache. The MOCS value must not be set to cache the data in L3.</p> <p>When clear, index data fetches bypass the L3 logic, therefore precluding the coherency of that data in the L3 cache. If the vertex buffer data can be cached in L3, the L3 cache must first be flushed to maintain vertex buffer data coherency.</p>	
	10	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	9:8	<b>Index Format</b>	
Format:		U2	
This field specifies the data format of the index buffer. All index values are UNSIGNED.			
<b>Value</b>		<b>Name</b>	
0h		BYTE	
1h	WORD		
2h	DWORD		
7	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	



<b>3DSTATE_INDEX_BUFFER_BODY</b>															
	<table border="1"> <tr> <td style="text-align: center;">6:0</td> <td><b>Memory Object Control State</b></td> </tr> <tr> <td>Format:</td> <td><b>MEMORY_OBJECT_CONTROL_STATE</b></td> </tr> <tr> <td colspan="2">Specifies the memory object control state for this index buffer.</td> </tr> </table>	6:0	<b>Memory Object Control State</b>	Format:	<b>MEMORY_OBJECT_CONTROL_STATE</b>	Specifies the memory object control state for this index buffer.									
6:0	<b>Memory Object Control State</b>														
Format:	<b>MEMORY_OBJECT_CONTROL_STATE</b>														
Specifies the memory object control state for this index buffer.															
1..2	<table border="1"> <tr> <td style="text-align: center;">63:0</td> <td><b>Buffer Starting Address</b></td> </tr> <tr> <td>Format:</td> <td>GraphicsAddress[63:0]</td> </tr> <tr> <td colspan="2" style="text-align: center;"><b>Description</b></td> </tr> <tr> <td colspan="2">VIRTUAL_ADDR[63:48] are ignored by the HW.</td> </tr> <tr> <td colspan="2">This field contains the size-aligned (as specified by Index Format) Graphics Address of the first element of interest within the index buffer. Software must program this value with the combination (sum) of the base address of the memory resource and the byte offset from the base address to the starting structure within the buffer.</td> </tr> <tr> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="2">Index Buffers can only be allocated in linear (not tiled) graphics memory.</td> </tr> </table>	63:0	<b>Buffer Starting Address</b>	Format:	GraphicsAddress[63:0]	<b>Description</b>		VIRTUAL_ADDR[63:48] are ignored by the HW.		This field contains the size-aligned (as specified by Index Format) Graphics Address of the first element of interest within the index buffer. Software must program this value with the combination (sum) of the base address of the memory resource and the byte offset from the base address to the starting structure within the buffer.		<b>Programming Notes</b>		Index Buffers can only be allocated in linear (not tiled) graphics memory.	
63:0	<b>Buffer Starting Address</b>														
Format:	GraphicsAddress[63:0]														
<b>Description</b>															
VIRTUAL_ADDR[63:48] are ignored by the HW.															
This field contains the size-aligned (as specified by Index Format) Graphics Address of the first element of interest within the index buffer. Software must program this value with the combination (sum) of the base address of the memory resource and the byte offset from the base address to the starting structure within the buffer.															
<b>Programming Notes</b>															
Index Buffers can only be allocated in linear (not tiled) graphics memory.															
3	<table border="1"> <tr> <td style="text-align: center;">31:0</td> <td><b>Buffer Size</b></td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> <tr> <td colspan="2">This field specifies the size of the buffer in bytes. Index accesses which straddle or go past the end of the buffer will return 0..Note that BufferSize=0 indicates that there is no valid data in the buffer.</td> </tr> <tr> <td colspan="2" style="text-align: center;"><b>Value</b></td> </tr> <tr> <td colspan="2">[0, FFFFFFFFh]</td> </tr> <tr> <td colspan="2" style="text-align: center;"><b>Name</b></td> </tr> <tr> <td colspan="2"></td> </tr> </table>	31:0	<b>Buffer Size</b>	Format:	U32	This field specifies the size of the buffer in bytes. Index accesses which straddle or go past the end of the buffer will return 0..Note that BufferSize=0 indicates that there is no valid data in the buffer.		<b>Value</b>		[0, FFFFFFFFh]		<b>Name</b>			
31:0	<b>Buffer Size</b>														
Format:	U32														
This field specifies the size of the buffer in bytes. Index accesses which straddle or go past the end of the buffer will return 0..Note that BufferSize=0 indicates that there is no valid data in the buffer.															
<b>Value</b>															
[0, FFFFFFFFh]															
<b>Name</b>															

## 3DSTATE\_MULTISAMPLE\_BODY

3DSTATE_MULTISAMPLE_BODY				
Source:	RenderCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:6	<b>Reserved</b>		
		Access: RO		
		Format: MBZ		
5	5	<b>Pixel Position Offset Enable</b>		
		Format: Enable		
		<p>Enables the device to offset pixel positions by 0.5 both in horizontal and vertical directions.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>Setting this field along with setting the Pixel Location to upper left and number of multisamples to greater than one will cause the device to offset pixel positions by 0.5 both in horizontal and vertical directions.</p> <p>It is to be noted this is done to adjust the pixel co-ordinate system to DX9 like, so any WM_HZ_OP screen space rectangles (eg: legacy HiZ Clear, Resolve etc) generated internally by driver in this mode needs to be aware of this offset adjustment and send the rectangles according to alignment restriction taking this offset adjustment into consideration.</p> <p>SW can choose to set this bit only for DX9 API. DX10/OpenGL API's should not have any effect by setting or not setting this bit.</p>		
4	4	<b>Pixel Location</b>		
		Format: U1		
		This field specifies where the device evaluates "pixel" (vs. centroid or sample) values/attributes.		
		<b>Value</b>	<b>Name</b>	<b>Description</b>
		0h	CENTER	Use the pixel center (0.5, 0.5 offset)
		1h	UL_CORNER	Use the pixel upper-left corner
<b>Programming Notes</b>				
The programming of this field is assumed to be a function of the API being supported. Specifically, it is expected that OpenGL and DX10+ APIs require CENTER selection, while DX9-APIs require UL_CORNER selection.				
When 3DSTATE_RASTER:: <b>ForcedSampleCount</b> is other than NUMRASTSAMPLES_0, this field must be 0h.				
3:1	3:1	<b>Number of Multisamples</b>		
		Format: U3		
This field specifies how many samples/pixel exist in all RTs and the Depth Buffer, as $\log_2(\#samples)$ . This field is valid regardless of the setting of <b>Multisample Rasterization Mode</b> .				

<b>3DSTATE_MULTISAMPLE_BODY</b>			
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	1	1 sample/pixel
	1h	2	2 samples/pixel
	2h	4	4 samples/pixel
	3h	8	8 samples/pixel
	4h	16	16 samples/pixel
	5h-7h	Reserved	
	<b>Programming Notes</b>		
	The setting of this field must match the <b>Number of Multisamples</b> field in SURFACE_STATE of all bound render targets.		
<b>0</b>	<b>Reserved</b>		
	Access:		RO
	Format:		MBZ

## 3DSTATE\_PRIMITIVE\_REPLICATION\_BODY

3DSTATE_PRIMITIVE_REPLICATION_BODY					
Source:	RenderCS				
Size (in bits):	160				
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000				
DWord	Bit	Description			
0	31:16	<b>Replica Mask</b> Specifies which replicas should be drawn. If bit $k$ ( $0 \leq k < 16$ ) is clear, then replica $k$ will not be rasterized. Bits $k \geq \text{numReplicas}$ are ignored			
	15:4	<b>Reserved</b> Access: RO Format: MBZ			
		<b>Replication Count</b> Format: U4 Specifies the number of replica positions produced by the last pre-raster shader. This value must match the SV_Position array length of the last pre-raster shader			
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0h, Fh]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0h, Fh]
Value	Name				
[0h, Fh]					
1	31:28	<b>Viewport Offset[7]</b> Format: U4 Specifies an offset to add to SV_ViewportArrayIndex for each replica			
	27:24	<b>Viewport Offset[6]</b> Format: U4 Specifies an offset to add to SV_ViewportArrayIndex for each replica			
	23:20	<b>Viewport Offset[5]</b> Format: U4 Specifies an offset to add to SV_ViewportArrayIndex for each replica			
	19:16	<b>Viewport Offset[4]</b> Format: U4 Specifies an offset to add to SV_ViewportArrayIndex for each replica			
	15:12	<b>Viewport Offset[3]</b> Format: U4 Specifies an offset to add to SV_ViewportArrayIndex for each replica			

<b>3DSTATE_PRIMITIVE_REPLICATION_BODY</b>			
	<p>11:8 <b>Viewport Offset[2]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4
	Format:	U4	
	<p>7:4 <b>Viewport Offset[1]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4
Format:	U4		
<p>3:0 <b>Viewport Offset[0]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4	
Format:	U4		
2	<p>31:28 <b>Viewport Offset[15]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4
	Format:	U4	
	<p>27:24 <b>Viewport Offset[14]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4
	Format:	U4	
	<p>23:20 <b>Viewport Offset[13]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4
	Format:	U4	
	<p>19:16 <b>Viewport Offset[12]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4
	Format:	U4	
<p>15:12 <b>Viewport Offset[11]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4	
Format:	U4		
<p>11:8 <b>Viewport Offset[10]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4	
Format:	U4		
<p>7:4 <b>Viewport Offset[9]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4	
Format:	U4		

<b>3DSTATE_PRIMITIVE_REPLICATION_BODY</b>				
	3:0	<p><b>Viewport Offset[8]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_ViewportArrayIndex for each replica</p>	Format:	U4
Format:	U4			
3	31:28	<p><b>RTAI Offset[7]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
	Format:	U4		
	27:24	<p><b>RTAI Offset[6]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
	Format:	U4		
	23:20	<p><b>RTAI Offset[5]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
	Format:	U4		
	19:16	<p><b>RTAI Offset[4]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
	Format:	U4		
	15:12	<p><b>RTAI Offset[3]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
Format:	U4			
11:8	<p><b>RTAI Offset[2]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4	
Format:	U4			
7:4	<p><b>RTAI Offset[1]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4	
Format:	U4			
3:0	<p><b>RTAI Offset[0]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4	
Format:	U4			
4	31:28	<p><b>RTAI Offset[15]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
Format:	U4			

<b>3DSTATE_PRIMITIVE_REPLICATION_BODY</b>			
27:24	<b>RTAI Offset[14]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
Format:	U4		
23:20	<b>RTAI Offset[13]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
Format:	U4		
19:16	<b>RTAI Offset[12]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
Format:	U4		
15:12	<b>RTAI Offset[11]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
Format:	U4		
11:8	<b>RTAI Offset[10]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
Format:	U4		
7:4	<b>RTAI Offset[9]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
Format:	U4		
3:0	<b>RTAI Offset[8]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U4</td> </tr> </table> <p>Specifies an offset to add to SV_RenderTargetArrayIndex for each replica</p>	Format:	U4
Format:	U4		

## 3DSTATE\_PS\_BLEND\_BODY

3DSTATE_PS_BLEND_BODY		
Source:	RenderCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31	<b>Alpha To Coverage Enable</b> Format: <input type="checkbox"/> Enable If set, indicates that AlphaToCoverage is on RT[0], since this bit must be set the same for all RTs in the MRT case.
	30	<b>Has Writeable RT</b> Format: <input type="checkbox"/> Enable When set indicates the there is at least one non-null RT w/ at least one channel write enabled
	29	<b>Color Buffer Blend Enable</b> Format: <input type="checkbox"/> Enable When set indicates that RT[0] has color buffer blend enabled.
	28:24	<b>Source Alpha Blend Factor</b> Format: <input type="text"/> <b>3D_Color_Buffer_Blend_Factor</b> Indicates the "source factor" in alpha Color Buffer Blending stage for RT[0]
	23:19	<b>Destination Alpha Blend Factor</b> Format: <input type="text"/> <b>3D_Color_Buffer_Blend_Factor</b> Indicates the "destination factor" in alpha Color Buffer Blending stage for RT[0]
	18:14	<b>Source Blend Factor</b> Format: <input type="text"/> <b>3D_Color_Buffer_Blend_Factor</b> Indicates the "source factor" in Color Buffer Blending stage for RT[0]
	13:9	<b>Destination Blend Factor</b> Format: <input type="text"/> <b>3D_Color_Buffer_Blend_Factor</b> Indicates the "destination factor" in Color Buffer Blending stage for RT[0]
	8	<b>Alpha Test Enable</b> Format: <input type="checkbox"/> Enable Indicates the AlphaTestEnable for RT[0]



<b>3DSTATE_PS_BLEND_BODY</b>					
7	<p><b>Independent Alpha Blend Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Indicates the Independent Alpha Blend Enable for RT[0] When enabled, the other fields in this instruction control the combination of the alpha components in the Color Buffer Blend stage. When disabled, the alpha components are combined in the same fashion as the color components.</p>	Format:	Enable		
Format:	Enable				
6:0	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				

## 3DSTATE\_PS\_BODY

3DSTATE_PS_BODY																
Source:	RenderCS															
Size (in bits):	352															
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000															
DWord	Bit	Description														
0..1	63:6	<p><b>Kernel Start Pointer 0</b></p> <table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[63:6]</td> </tr> </table> <p>Specifies the 64-byte aligned address offset of the first instruction in the kernel[0]. This pointer is relative to the <b>Instruction Base Address</b>.</p>	Format:	InstructionBaseOffset[63:6]												
	Format:	InstructionBaseOffset[63:6]														
5:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ											
Access:	RO															
Format:	MBZ															
2	31	<p><b>Single Program Flow</b></p> <p>Single Program Flow (SPF) specifies the initial condition of the kernel program as either a single program flow (SIMDn<sub>xm</sub> with m = 1) or as multiple program flows (SIMDn<sub>xm</sub> with m &gt; 1). See CR0 description in ISA Execution Environment.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Multiple</td> <td>Multiple Program Flows</td> </tr> <tr> <td>1h</td> <td>Single</td> <td>Single Program Flows</td> </tr> </tbody> </table>	Value	Name	Description	0h	Multiple	Multiple Program Flows	1h	Single	Single Program Flows					
	Value	Name	Description													
0h	Multiple	Multiple Program Flows														
1h	Single	Single Program Flows														
30	<p><b>Vector Mask Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>When SPF=0, Vector Mask Enable (VME) specifies which mask to use to initialize the initial channel enables. When SPF=1, VME specifies which mask to use to generate execution channel enables.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Dmask</td> <td>Channels are enabled based on the dispatch mask</td> </tr> <tr> <td>1h</td> <td>Vmask</td> <td>Channels are enabled based on the vector mask</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0h	Dmask	Channels are enabled based on the dispatch mask	1h	Vmask	Channels are enabled based on the vector mask				
Format:	Enable															
Value	Name	Description														
0h	Dmask	Channels are enabled based on the dispatch mask														
1h	Vmask	Channels are enabled based on the vector mask														
29:27	<p><b>Sampler Count</b></p> <p>Specifies how many samplers (in multiples of 4) the vertex shader 0 kernel uses. Used only for prefetching the associated sampler state entries.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,4]</td> <td></td> <td></td> </tr> <tr> <td>0h</td> <td>No Samplers</td> <td>no samplers used</td> </tr> <tr> <td>1h</td> <td>1-4 Samplers</td> <td>between 1 and 4 samplers used</td> </tr> <tr> <td>2h</td> <td>5-8 Samplers</td> <td>between 5 and 8 samplers used</td> </tr> </tbody> </table>	Value	Name	Description	[0,4]			0h	No Samplers	no samplers used	1h	1-4 Samplers	between 1 and 4 samplers used	2h	5-8 Samplers	between 5 and 8 samplers used
	Value	Name	Description													
[0,4]																
0h	No Samplers	no samplers used														
1h	1-4 Samplers	between 1 and 4 samplers used														
2h	5-8 Samplers	between 5 and 8 samplers used														

## 3DSTATE\_PS\_BODY

	3h	9-12 Samplers	between 9 and 12 samplers used
	4h	13-16 Samplers	between 13 and 16 samplers used
	5h-7h		Reserved
26	<b>Single Precision Denormal Mode</b> Specifies the single precision denormal mode used by the dispatched thread.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	Flushed to Zero	Single Precision denormals are flushed to zero
	1h	Retained	Single Precision denormals are retained
25:18	<b>Binding Table Entry Count</b> Specifies how many binding table entries the kernel uses. Used only for prefetching of the binding table entries and associated surface state. <b>Note:</b> For kernels using a large number of binding table entries, it may be advantageous to set this field to zero to avoid prefetching too many entries and thrashing the state cache. This field is ignored if [PS Function Enable] is DISABLED.  When [HW Generated Binding Table] bit is enabled: This field indicates which cache lines (512bit units - 32 Binding Table Entry section) should be fetched. Each bit in this field corresponds to a cache line. Only the 1st 4 non-zero Binding Table entries of each 32 Binding Table entry section prefetched will have its surface state prefetched. See 3D Pipeline for more information.		
	<b>Programming Notes</b>		
	When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.		
17	<b>Thread Dispatch Priority</b> Specifies the priority of the thread for dispatch.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	Normal	Normal Priority
	1h	High	High Priority
16	<b>Floating Point Mode</b> Specifies the floating point mode used by the dispatched thread.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	IEEE-754	Use IEEE-754 rules
	1h	Alternate	Use alternate rules
15:14	<b>Rounding Mode</b> Specifies the rounding mode used by the dispatched thread.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	RTNE	Round to Nearest Even
	1h	RU	Round toward +infinity
	2h	RD	Round toward -infinity

<b>3DSTATE_PS_BODY</b>			
	3h	RTZ	Round toward zero
	13	<b>Illegal Opcode Exception Enable</b>	
		Format:	Enable
		This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.	
	12	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	11	<b>Mask Stack Exception Enable</b>	
		Format:	Enable
	This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment.		
	10:8	<b>Reserved</b>	
		Format:	MBZ
	7	<b>Software Exception Enable</b>	
		Format:	Enable
		This bit gets loaded into EU CR0.1[13] (note the bit # difference). See Exceptions and ISA Execution Environment.	
	6:0	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
3..4	63:32	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	31:10	<b>Scratch Space Base Pointer</b>	
		Format:	GeneralStateOffset[31:10]
	Specifies the 1k-byte aligned address offset to scratch space for use by the kernel. This pointer is relative to the General State Base Address.		
	<b>Programming Notes</b>		
	Scratch Space per slice is computed based on 4 sub-slices. SW must allocate scratch space enough so that each slice has 4 slices allowed.		
	9:4	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ

<b>3DSTATE_PS_BODY</b>								
	3:0	<p><b>Per Thread Scratch Space</b></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 each thread. The driver must allocate enough contiguous scratch space, pointed to by the Scratch Space Pointer, to ensure that the Maximum Number of Threads each get Per Thread Scratch Space size without exceeding the driver-allocated scratch space.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td>indicating [1k bytes, 2M bytes] in powers of two</td> </tr> </tbody> </table>	Format:	U4	Value	Name	[0,11]	indicating [1k bytes, 2M bytes] in powers of two
	Format:	U4						
	Value	Name						
[0,11]	indicating [1k bytes, 2M bytes] in powers of two							
5	31:23	<p><b>Maximum Number of Fused Threads Per PSD</b></p> <table border="1"> <tr> <td>Format:</td> <td>U9-1</td> </tr> </table> <p style="text-align: center;"><b>Description</b></p> <p>Specifies the maximum number of simultaneous virtual fused threads allowed to be active per Pixel Shader Dispatch(PSD). PSD serves a pair of subslices. This bit-field can be programmed in the range: [0,63] each integer in the range linearly maps to maximum number of virtual fused threads in the range: [1, 64]. The allowable range is larger than the maximum number of fused physical threads per PSD. It is advantageous for performance reasons to allow more virtual threads than physical threads to ensure maximum usage of compute resources. Each fused thread represents 2 threads.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>If this field is changed between 3DPRIMITIVE commands, a PIPE_CONTROL command with Stall at Pixel Scoreboard set is required to be issued.</p> <p>This note only applies to 2 pass AMFS approach where AMFS unit launches Texel Shaders. This deadlock workaround is not needed for 3 pass approach where evaluate message does not cause AMFS unit to spawn Texel Shaders.</p> <p>When Pixel Shader contains one or more evaluate message for Procedural Texture, and AMFS is expected to dispatch Texel Shaders, the maximum number of fused virtual threads must be programmed to be less than maximum number of fused physical threads possible per PSD. Maximum number of fused physical threads is device specific. (see device specific programming notes for this field)This ensures that AMFS unit never gets deadlocked by restricting PSD from using all available compute resources. For typical Procedural Texture usage model we can program one less than maximum physical fused threads.</p>	Format:	U9-1				
		Format:	U9-1					
		22	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
		Access:	RO					
Format:	MBZ							
21	<p><b>Pixel Scoreboard Disable</b></p> <p>Setting this bit disables the pixel shader scoreboard for ordering the RTs and ROVs on the same screen space coordinates.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>There are cases when ordering the render target outputs or ROV outputs from the shader, for example:</p>							

## 3DSTATE\_PS\_BODY

		<p>1) all blend functions are commutative, here are the most common cases: BLEND_OP = ADD or MIN or MAX and both src and dst blend factors are constants = 1.0.</p> <p>2) There is no over-draw in the render pass (for example full screen 3D PASS which accesses a pixel in the color buffer just once).</p> <p>When HW detects the change in this bit, it implicitly performs the PS scoreboard stall before allowing the subsequent group of pixel shader threads.</p> <p>When SIMD32 is enabled, this bit <b>must not be set</b>. Pixel Scoreboard must always be enabled when SIMD32 is enabled.</p>											
	20	<p><b>Reserved</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ						
Access:	RO												
Format:	MBZ												
	19:12	<p><b>Clear/Resolve BTI for Render Target</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Description</th> </tr> <tr> <td colspan="2"> <p>When Color Clear or Resolve bit is set in this state packet, this bit field indicates BTI to be used to access the Render Target Surface that's being cleared/resolved.</p> </td> </tr> </table>		Description		<p>When Color Clear or Resolve bit is set in this state packet, this bit field indicates BTI to be used to access the Render Target Surface that's being cleared/resolved.</p>							
Description													
<p>When Color Clear or Resolve bit is set in this state packet, this bit field indicates BTI to be used to access the Render Target Surface that's being cleared/resolved.</p>													
	11	<p><b>Push Constant Enable</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field must be enabled if the sum of the <b>PS Constant Buffer [3:0] Read Length</b> fields in 3DSTATE_CONSTANT_PS is nonzero, and must be disabled if the sum is zero.</p>		Format:	Enable								
Format:	Enable												
	10	<p><b>3D Scoreboard Address Size select</b> Select the granularity use for scoreboard address calculation.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; background-color: #e6f2ff;">Value</th> <th style="text-align: center; background-color: #e6f2ff;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">2x2</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">4x2</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2"> <p>This bit field allows programmable pixel scoreboard granularity: 2X2 pixel block(value = 0) or 4X2 pixel block(value = 1). When the value of this bit field changes, HW detects the change and takes the action to either force thread-group dependency or stalls at the scoreboard (based on the MMIO(PSS_MODE2):Thread Group Dependency Control).</p> <p>When enabling fused-SIMD32 dispatch mode, HW implicitly sets the scoreboard size to 4X2 independent of the value of this bit-field.</p> </td> </tr> </table>		Value	Name	0h	2x2	1h	4x2	Programming Notes		<p>This bit field allows programmable pixel scoreboard granularity: 2X2 pixel block(value = 0) or 4X2 pixel block(value = 1). When the value of this bit field changes, HW detects the change and takes the action to either force thread-group dependency or stalls at the scoreboard (based on the MMIO(PSS_MODE2):Thread Group Dependency Control).</p> <p>When enabling fused-SIMD32 dispatch mode, HW implicitly sets the scoreboard size to 4X2 independent of the value of this bit-field.</p>	
Value	Name												
0h	2x2												
1h	4x2												
Programming Notes													
<p>This bit field allows programmable pixel scoreboard granularity: 2X2 pixel block(value = 0) or 4X2 pixel block(value = 1). When the value of this bit field changes, HW detects the change and takes the action to either force thread-group dependency or stalls at the scoreboard (based on the MMIO(PSS_MODE2):Thread Group Dependency Control).</p> <p>When enabling fused-SIMD32 dispatch mode, HW implicitly sets the scoreboard size to 4X2 independent of the value of this bit-field.</p>													
	9	<p><b>Overlapping Subspans Enable</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field indicates if two subspans (from two objects) rasterized to same screen-space XY coordinates can be packed into a single EU thread payload or across fused threads.</p> <p>The shader compiler must set this field to DISABLED when pixel shader code requires serialized execution on per-pixel basis; examples include pixel shader using RT reads or pixel sync.</p> <p>This field must be set to DISABLED if kernel has any coding structures that can create possibility</p>		Format:	Enable								
Format:	Enable												

## 3DSTATE\_PS\_BODY

of younger object (e.g. upper fused thread or upper SIMD8 of dual-SIMD8 pair) to issue message before older object in fused thread.

Examples include:

- SIMD16/dual-SIMD8 thread issuing 2 SIMD8 messages
- A message issued from within if-else. (if message else different message)
- 3DSTATE\_PS\_EXTRA:killpix is set and depth/stencil write is enabled
- 3DSTATE\_PS\_EXTRA:computed depth/stencil is set and depth/stencil write is enabled
- Read and write to same UAV or RT

### Programming Notes

When 3DSTATE\_PS:Pixel Scoreboard Disable is set, this field **must be set**. The intent of scoreboard disable is to allow overlapping primitives in the shader stage and therefore disabling overlapping pixels defeats the purpose of this optimization.

### 8 **Render Target Fast Clear Enable**

Format:	Enable
---------	--------

This field is set to enable fast clear of the bound render targets. See "**Render Target Fast Clear**" and "**MCS/CCS Buffers for Render Targets**" for restrictions on enabling this field.

A general programming sequence for doing the Render Target Fast Clear requires:

- setting up the Render Target State with RT that needs to be cleared as well as clear value is stored at the Clear Value Address in the RT State
- Provide the BTI for that RT in this state packet
- set this bit in the state packet provided the fast clear guidelines described in the Fast Clear section
- DRAW command with a rectangle (scaled appropriately) as a primitive

### Programming Notes

For PoSH based Tiled Rendering, Color Fast clear operation is recommended to be performed outside of tile pass, for performance reasons. After Fast clear, render cache flush is required.

For PoSH based Tiled Rendering, Color Fast clear can be inside the tile pass without significant performance penalty and it does not require render cache flush after fast clear of color.

When this bit is set, corresponding BTI for the render target that is being cleared must be equal to 0.

When this bit is set, RENDER\_SURFACE\_STATE type must not be NULL.

3D/Volumetric surfaces do not support Fast Clear operation.

### 7:6 **Render Target Resolve Type**

Format:	U2
---------	----

Specifies what type of Render Target Resolve is needed for the surface to be consumed properly by the end Client. Programming notes below.

Value	Name	Description	Programming Notes
0h	RESOLVE_DISABLED	No Resolve Needed	
1h	RESOLVE_PARTIAL	Partial resolve is for resolving RT for clear values i.e. it leaves	Display engine does not support unresolved clear values in the

## 3DSTATE\_PS\_BODY

			no cache lines at implied clear value.	display buffer, hence this resolve is required before binding any compressed RT to the display via flip commands.
	2h	FAST_CLEAR_0	Fast Clear to 0 during Clear Pass; Used to Initialize CCS Buffer with 0s to support Lossless Compressed Without Clear.	This state has to be programmed only with Render Target Fast Clear Enable described above. If the Render Target Fast Clear = 0, this Field Cannot be programmed to 2h.
	3h	RESOLVE_FULL	Full Resolve is for Resolving RT for Clear/Compressed to Uncompressed State	
<b>Programming Notes</b>				
When this bit is set, corresponding BTI for the render target that is being resolved must be equal to 0.				
When this bit is set, RENDER_SURFACE_STATE type must not be NULL.				
For multisample render target, this field must be RESOLVE_DISABLED.				
3D/Volumetric surfaces do not support in-place resolve pass operation.				
5	<b>Dual-SIMD8 Dispatch Enabled</b>			
	Format:		Enable	
This field determines type of pixel shader enabled by <b>8 Pixel Dispatch or Dual-8 Pixel Dispatch Enable</b> field.				
If DISABLED, the pixel shader kernel receives SIMD8 payload (8 pixels from 1 object).				
If ENABLED, the pixel shader kernel receives dual-SIMD8 payload (8 pixels from 1 <sup>st</sup> object and 8 pixels from 2 <sup>nd</sup> object)				
4:3	<b>Position XY Offset Select</b>			
	Format:		U2	
This field specifies if/what Position XY Offset values are passed in the PS payload. Note that these are per-slot (pixel sample) offsets, and therefore separate from the subspan XY coordinates passed in R1.				
	<b>Value</b>	<b>Name</b>	<b>Description</b>	
	0h	POSOFFSET_NONE	No Position XY Offsets are included in the PS payload.	
	1h	Reserved		
	2h	POSOFFSET_CENTROID	Position XY Offsets will be passed in the PS payload, and these will reflect the Centroid position(s).	
	3h	POSOFFSET_SAMPLE	Position XY Offsets will be passed in the PS payload, and these will reflect the multisample position(s).	



## 3DSTATE\_PS\_BODY

Programming Notes															
SW Recommendation: If the PS kernel needs the Position Offsets to compute a Position XY value, this field should match Position ZW Interpolation Mode to ensure a consistent position.xyzw computation															
If the PS kernel does not need the Position XY Offsets to compute a Position Value, then this field should be programmed to POSOFFSET_NONE, as the PS kernel should be using the various barycentric inputs to evaluate other-than-position attributes. However, this field can be used to pass Centroid or Sample offsets in the payload for special test modes (e.g., where barycentric coordinates are computed in the PS vs. being HW-generated and passed in the payload).															
MSDISPMODE_PERSAMPLE is required in order to select POSOFFSET_SAMPLE.															
2	<b>32 Pixel Dispatch Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>Enables the Windower to dispatch 8 subspans in one payload. Variable Pixel Dispatch in Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.</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">When NUM_MULTISAMPLES = 16 or FORCE_SAMPLE_COUNT = 16, SIMD32 Dispatch must not be enabled for PER_PIXEL dispatch mode.</td> </tr> <tr> <td colspan="2">Must not be enabled when dispatch rate is sample AND NUM_MULTISAMPLES &gt; 1. SIMD32 may only be enabled if SIMD16 or (dual)SIMD8 is also enabled.</td> </tr> <tr> <td colspan="2">Must not be enabled when dispatch rate is coarse.</td> </tr> </tbody> </table>	Format:	Enable	Programming Notes		When NUM_MULTISAMPLES = 16 or FORCE_SAMPLE_COUNT = 16, SIMD32 Dispatch must not be enabled for PER_PIXEL dispatch mode.		Must not be enabled when dispatch rate is sample AND NUM_MULTISAMPLES > 1. SIMD32 may only be enabled if SIMD16 or (dual)SIMD8 is also enabled.		Must not be enabled when dispatch rate is coarse.					
Format:	Enable														
Programming Notes															
When NUM_MULTISAMPLES = 16 or FORCE_SAMPLE_COUNT = 16, SIMD32 Dispatch must not be enabled for PER_PIXEL dispatch mode.															
Must not be enabled when dispatch rate is sample AND NUM_MULTISAMPLES > 1. SIMD32 may only be enabled if SIMD16 or (dual)SIMD8 is also enabled.															
Must not be enabled when dispatch rate is coarse.															
1	<b>16 Pixel Dispatch Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>Enables the Windower to dispatch 4 subspans in one payload. Variable Pixel Dispatch in Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.</p>	Format:	Enable												
Format:	Enable														
0	<b>8 Pixel Dispatch Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">Enables the Windower to dispatch 2 subspans from 1 object (polygon) in one payload. Variable Pixel Dispatch in Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.</td> </tr> <tr> <td colspan="2">If Dual-SIMD8 Dispatch Enabled, kernel pointer referenced by this field isDual-SIMD8 kernel pointer instead of SIMD8 kernel pointer.Dual-SIMD8 and SIMD8 modes are mutually exclusive and use the same kernel pointer entry.</td> </tr> <tr> <td colspan="2">If Dual-SIMD8 Dispatch Enabled, the Widower packs 2 subspans from one object followed by 2 subspans from another object into one PS thread payload.</td> </tr> </tbody> </table> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">When Render Target Fast Clear Enable is ENABLED or Render Target Resolve Type =</td> </tr> </tbody> </table>	Format:	Enable	Description		Enables the Windower to dispatch 2 subspans from 1 object (polygon) in one payload. Variable Pixel Dispatch in Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.		If Dual-SIMD8 Dispatch Enabled, kernel pointer referenced by this field isDual-SIMD8 kernel pointer instead of SIMD8 kernel pointer.Dual-SIMD8 and SIMD8 modes are mutually exclusive and use the same kernel pointer entry.		If Dual-SIMD8 Dispatch Enabled, the Widower packs 2 subspans from one object followed by 2 subspans from another object into one PS thread payload.		Programming Notes		When Render Target Fast Clear Enable is ENABLED or Render Target Resolve Type =	
Format:	Enable														
Description															
Enables the Windower to dispatch 2 subspans from 1 object (polygon) in one payload. Variable Pixel Dispatch in Section: Pixel Grouping (Dispatch size) control for valid pixel dispatch combinations.															
If Dual-SIMD8 Dispatch Enabled, kernel pointer referenced by this field isDual-SIMD8 kernel pointer instead of SIMD8 kernel pointer.Dual-SIMD8 and SIMD8 modes are mutually exclusive and use the same kernel pointer entry.															
If Dual-SIMD8 Dispatch Enabled, the Widower packs 2 subspans from one object followed by 2 subspans from another object into one PS thread payload.															
Programming Notes															
When Render Target Fast Clear Enable is ENABLED or Render Target Resolve Type =															

		<b>3DSTATE_PS_BODY</b>	
		RESOLVE_PARTIAL or RESOLVE_FULL, this bit must be DISABLED.	
6	31:23	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	22:16	<b>Dispatch GRF Start Register For Constant/Setup Data 0</b>	
		Format:	U7
		Specifies the starting GRF register number for the Constant/Setup portion of the thread payload for kernel[0].	
		<b>Value</b>	<b>Name</b>
		[0,127]	
	15	<b>Reserved</b>	
		Access:	RO
Format:		MBZ	
14:8	<b>Dispatch GRF Start Register For Constant/Setup Data 1</b>		
	Format:	U7	
	Specifies the starting GRF register number for the Constant/Setup portion of the thread payload for kernel[1].		
	<b>Value</b>	<b>Name</b>	
	[0,127]		
7	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
6:0	<b>Dispatch GRF Start Register For Constant/Setup Data 2</b>		
	Format:	U7	
	Specifies the starting GRF register number for the Constant/Setup portion of the thread payload for kernel[2].		
	<b>Value</b>	<b>Name</b>	
	[0,127]		
7..8	63:6	<b>Kernel Start Pointer 1</b>	
		Format:	InstructionBaseOffset[63:6]
	Specifies the 64-byte aligned address offset of the first instruction in kernel[1]. This pointer is relative to the Instruction Base Address.		
5:0	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	

<b>3DSTATE_PS_BODY</b>				
9..10	63:6	<b>Kernel Start Pointer 2</b>		
		<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>InstructionBaseOffset[63:6]</td> </tr> </table> <p>Specifies the 64-byte aligned address offset of the first instruction in kernel[2]. This pointer is relative to the <b>Instruction Base Address</b>.</p>	Format:	InstructionBaseOffset[63:6]
	Format:	InstructionBaseOffset[63:6]		
	5:0	<b>Reserved</b>		
<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:
Access:	RO			
Format:	MBZ			

## 3DSTATE\_PS\_EXTRA\_BODY

3DSTATE_PS_EXTRA_BODY										
Source:	RenderCS									
Size (in bits):	32									
Default Value:	0x00000000									
DWord	Bit	Description								
0	31	<b>Pixel Shader Valid</b> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>When set indicates a valid pixel shader. When this bit clear the rest of this command should also be clear.</p>	Format:	Enable						
	Format:	Enable								
	30	<b>Pixel Shader Does not write to RT</b> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>When set indicates the pixel shader does not write to render target.</p> <table border="1"> <thead> <tr> <th colspan="2">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">When Pixel Shader writes to UAV but does not write to RT, a dummy render target write is required to convey EOT to the PS dispatch function. Hence, this bit must be reset in this case. If there is no RT or a NULL RT, Pixel Shader Kills Pixel is reset, and there is no UAV output from PS, SW must set this bit.</td> </tr> <tr> <td colspan="2">When Pixel Shader has evaluate message present, i.e. '3DSTATE_PS_EXTRA:PS has Evaluate Message' is enabled, this bit field must be reset.</td> </tr> </tbody> </table>	Format:	Enable	Programming Notes		When Pixel Shader writes to UAV but does not write to RT, a dummy render target write is required to convey EOT to the PS dispatch function. Hence, this bit must be reset in this case. If there is no RT or a NULL RT, Pixel Shader Kills Pixel is reset, and there is no UAV output from PS, SW must set this bit.		When Pixel Shader has evaluate message present, i.e. '3DSTATE_PS_EXTRA:PS has Evaluate Message' is enabled, this bit field must be reset.	
	Format:	Enable								
	Programming Notes									
When Pixel Shader writes to UAV but does not write to RT, a dummy render target write is required to convey EOT to the PS dispatch function. Hence, this bit must be reset in this case. If there is no RT or a NULL RT, Pixel Shader Kills Pixel is reset, and there is no UAV output from PS, SW must set this bit.										
When Pixel Shader has evaluate message present, i.e. '3DSTATE_PS_EXTRA:PS has Evaluate Message' is enabled, this bit field must be reset.										
29	<b>oMask Present to Render Target</b> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit is inserted in the PS payload header and made available to the DataPort (either via the message header or via header bypass) to indicate that oMask data from the shader (one or two phases) is included in Render Target Write messages. If present, the oMask data is used to mask off samples.</p>	Format:	Enable							
Format:	Enable									
28	<b>Pixel Shader Kills Pixel</b> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the PS kernel has the ability to kill (discard) pixels or samples, other than due to depth or stencil testing. This bit is required to be ENABLED in the following situations:</p> <ul style="list-style-type: none"> <li>The API pixel shader program contains "killpix" or "discard" instructions, or other code in the pixel shader kernel that can cause the final pixel mask to differ from the pixel mask received on dispatch.</li> </ul>	Format:	Enable							
Format:	Enable									
27:26	<b>Pixel Shader Computed Depth Mode</b> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field specifies the computed depth mode for the pixel shader.</p>	Format:	U2							
Format:	U2									

## 3DSTATE\_PS\_EXTRA\_BODY

		Value	Name	Description		
		0h	PSCDEPTH_OFF	Pixel shader does not compute depth		
		1h	PSCDEPTH_ON	Pixel shader computes depth with no guarantee as to its value		
		2h	PSCDEPTH_ON_GE			
		3h	PSCDEPTH_ON_LE	Pixel shader computes depth and guarantees that oDepth <= SourceDepth If the Position ZW interpolation mode in 3DSTATE_WM does not match the DX Multisample Rasterization mode in 3DSTATE_RASTER, HW will internally convert to PSCDEPTH_ON.		
		<b>Programming Notes</b>				
		If this field is set to any value other than PSCDEPTH_OFF, a multi-phase shader (i.e. dispatch RATE_COARSE or RATE_PIXEL with pixel/sample loops or sample loop respectively) must output depth and render targets only at the last phase.				
		When PS dispatch rate is COARSE_RATE, this field must be programmed to PSCDEPTH_OFF.				
25	<b>Force Computed Depth</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Enable</td> </tr> </table>			Format:	Enable
Format:	Enable					
		<b>Programming Notes</b>				
		This field should be left DISABLED. This field should not be tested for functional validation.				
24	<b>Pixel Shader Uses Source Depth</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the PS kernel requires the source depth value (vPos.z) to be passed in the payload. The source depth value is interpolated according to the Position ZW Interpolation Mode state.</p>			Format:	Enable
Format:	Enable					
		<b>Programming Notes</b>				
		This bit cannot be enabled when dispatch rate is RATE_COARSE				
23	<b>Pixel Shader Uses Source W</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the PS kernel requires the interpolated source W value (vPos.w) to be passed in the payload. The W value is interpolated according to the Position ZW Interpolation Mode state.</p>			Format:	Enable
Format:	Enable					
22	<b>Pixel Shader Requires Requested Coarse Pixel Shading Size</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the PS kernel requires values of requested coarse pixel shading size to be passed in the payload for each 2x2 coarse pixel quad. Note: Actual coarse pixel shading rate is always delivered (constant across thread slot). This bit can only be set when dispatch rate is RATE_COARSE.</p>			Format:	Enable
Format:	Enable					

## 3DSTATE\_PS\_EXTRA\_BODY

21	<b>Pixel Shader Requires Source Depth and/or W Plane Coefficients</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the PS kernel requires the source depth and/or W plane equations to be passed in the payload. Note: both attributes are always delivered in same message phase, even if only one is used.</p>	Format:	Enable						
Format:	Enable									
20	<b>Pixel Shader Requires Perspective Bary Plane Coefficients</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the PS kernel requires the perspective plane coefficients to be passed in the payload.</p>	Format:	Enable						
Format:	Enable									
19	<b>Pixel Shader Requires Non-Perspective Bary Plane Coefficients</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the PS kernel requires the non-perspective plane coefficients to be passed in the payload.</p>	Format:	Enable						
Format:	Enable									
18	<b>Pixel Shader Requires Subpixel Sample Offsets</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the PS kernel requires the sub-pixel sample offsets to be passed in the payload.</p>	Format:	Enable						
Format:	Enable									
17	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ									
16:12	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ									
11	<b>PS has Evaluate Message</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p style="text-align: center; color: #0070C0; margin: 0;"><b>Description</b></p> <p style="margin: 0;">This bit indicates if Pixel Shader has Evaluate Message typically used in conjunction with AMFS.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #D9E1F2;"> <th style="text-align: center; padding: 5px;">Value</th> <th style="text-align: center; padding: 5px;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center; padding: 5px;">0</td> <td style="padding: 5px;">Disable <b>[Default]</b></td> </tr> <tr> <td style="text-align: center; padding: 5px;">1</td> <td style="padding: 5px;">Enable</td> </tr> </tbody> </table> <p style="text-align: center; color: #0070C0; margin-top: 5px;"><b>Programming Notes</b></p> <p style="margin: 0;">This bit must be reset if 3DSTATE_PS_EXTRA:PS Valid bit is not set.</p> <p style="margin: 0;">This bit must be reset if Cache Mode Register 1 [6] is set , ie Shader Independent AMFS is enabled</p> </div>	Format:	Enable	Value	Name	0	Disable <b>[Default]</b>	1	Enable
Format:	Enable									
Value	Name									
0	Disable <b>[Default]</b>									
1	Enable									

<b>3DSTATE_PS_EXTRA_BODY</b>					
10:9	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO			
Format:	MBZ				
8	<b>Attribute Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Description</th> </tr> </table> <p>This field must be enabled if the Number of SF Output Attributes field in 3DSTATE_SBE is nonzero, and must be disabled if that field is zero.</p>	Format:	Enable	Description	
Format:	Enable				
Description					
7	<b>Pixel Shader Disables Alpha To Coverage</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>When set indicates the pixel shader AlphaToCoverage should be disabled due to oMask output. The setting of this bit is API dependent.</p>	Format:	Enable		
Format:	Enable				
6	<b>Pixel Shader Is Per Sample</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit, when ENABLED, indicates that the pixel shader is dispatched at the per sample shading rate. If this bit is DISABLED, the dispatch rate is determined by the value of Pixel Shader Is Per Coarse Pixel. If this bit is ENABLED, Pixel Shader Is Per Coarse Pixel bit must be DISABLED.</p>	Format:	Enable		
Format:	Enable				
5	<b>Pixel Shader Computes Stencil</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field when set indicates that the pixel shader computes the stencil reference value.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </table> <p>If this field is ENABLED, a multi-phase shader (i.e. dispatch RATE_COARSE or RATE_PIXEL with pixel/sample loops or sample loop respectively) must output stencil and render targets only at the last phase.</p> <p>When Pixel Shader is at COARSE_RATE, this field must not be set.</p>	Format:	Enable	Programming Notes	
Format:	Enable				
Programming Notes					
4	<b>Pixel Shader Is Per Coarse Pixel</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>If Pixel Shader Is Per Sample is DISABLED and this bit is ENABLED, the pixel shader is dispatched at the per coarse pixel shading rate. If Pixel Shader Is Per Sample is DISABLED and this bit is DISABLED, the pixel shader is dispatched at the per pixel shading rate. If Pixel Shader Is Per Sample is ENABLED, this bit must be DISABLED.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Restriction</th> </tr> </table> <p>SIMD32 kernel version cannot be configured when this bit is ENABLED.</p>	Format:	Enable	Restriction	
Format:	Enable				
Restriction					
3	<b>Pixel Shader Pulls Bary</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit indicates if Pixel Shader uses Pull Bary i.e. PI message. If this bit is reset, PS does not do Pull Bary.</p>	Format:	Enable		
Format:	Enable				

## 3DSTATE\_PS\_EXTRA\_BODY

2	<b>Pixel Shader Has UAV</b>			
	Format:	Enable		
	Format:	U1 Enumerated Type		
	This field when set indicates that the pixel shader has a UAV attached to it.			
1:0	<b>Input Coverage Mask State</b>			
	Format:	U2		
	This field indicates the type of input coverage mask that the PS kernel requires to be passed in the payload.			
	<b>Value</b>	<b>Name</b>	<b>Description</b>	<b>Programming Notes</b>
	0h	NONE	Pixel shader does not use input coverage masks.	
	1h	NORMAL	Input Coverage masks based on outer conservatism and factors in SAMPLE_MASKs. If Pixel is conservatively covered, all samples are enabled.	
	2h	INNER_CONSERVATIVE	Input Coverage masks based on inner conservatism. If Pixel is conservatively fully covered all samples are enabled else none of the samples are covered.	<p>When PS Dispatch Rate is at Coarse Pixel, requesting this Input Coverage Mask mode is illegal and not supported by HW.</p> <p>Input coverage masks based on inner conservatism incorrectly ANDs SAMPLE_MASK in HW. Therefore, PS must retrieve the INNER coverage mask per pixel by bit-wise OR operation.</p> <p>☐ Input coverage masks based on inner conservatism incorrectly ANDs SAMPLE_MASK in HW. Therefore, PS must retrieve the INNER coverage mask per pixel by bit-wise OR operation.</p> <p>☐ Input coverage masks based on inner conservatism incorrectly ANDs SAMPLE_MASK in HW. Therefore, PS must retrieve the INNER coverage mask</p>



<b>3DSTATE_PS_EXTRA_BODY</b>			
			per pixel by bit-wise OR operation.
	3h	DEPTH_COVERAGE	Input coverage masks are computed after factoring depth/stencil test results, only if Early Depth Stencil Test is enabled. If Early Depth Stencil Test is not enabled, HW uses NORMAL Input Coverage Masks.



## 3DSTATE\_PTBR\_MARKER\_BODY

3DSTATE_PTBR_MARKER_BODY		
Source:	RenderCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:2	<b>Reserved</b>
		Access: RO
		Format: MBZ
	1	<b>End of Tile</b>
		Format: Enable When set, indicates marker stating End of Tile in the command sequence.
	0	<b>Start of Tile</b>
Format: Enable When set, indicates marker stating Start of Tile in the command sequence.		

## 3DSTATE\_PTBR\_TILE\_SELECT\_BODY

3DSTATE_PTBR_TILE_SELECT_BODY									
Source:	RenderCS								
Size (in bits):	32								
Default Value:	0x00000000								
DWord	Bit	Description							
0	31	<b>Free Render List Disable</b>							
		Format: Disable							
		This bit controls the recycling (Freeing up, add back to the free pool) of the visibility data pages by render pipe.							
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Render pipe will free the pages to be recycled after consuming the visibility data for the current tile.</td> </tr> <tr> <td>1</td> <td></td> <td>Render pipe will not free the pages to be recycled after consuming the visibility data for the current tile.</td> </tr> </tbody> </table>	Value	Name	Description	0		Render pipe will free the pages to be recycled after consuming the visibility data for the current tile.	1
Value	Name	Description							
0		Render pipe will free the pages to be recycled after consuming the visibility data for the current tile.							
1		Render pipe will not free the pages to be recycled after consuming the visibility data for the current tile.							
30		<b>Geometry Statistics Disable</b>							
		Format: Disable							
		This bit controls the incrementing statistics counters in geometry units (VF, VS, HS, TE, TDS, GS, SOL, CL, SF).							
		<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>Geometry units (VF, VS, HS, TE, TDS, GS, SOL, CL, SF)will increment their pipeline statistics counters.</td> </tr> <tr> <td>1</td> <td></td> <td>Geometry units (VF, VS, HS, TE, TDS, GS, SOL, CL, SF)will not increment their pipeline statistics counters.</td> </tr> </tbody> </table>	Value	Name	Description	0	[Default]	Geometry units (VF, VS, HS, TE, TDS, GS, SOL, CL, SF)will increment their pipeline statistics counters.	1
Value	Name	Description							
0	[Default]	Geometry units (VF, VS, HS, TE, TDS, GS, SOL, CL, SF)will increment their pipeline statistics counters.							
1		Geometry units (VF, VS, HS, TE, TDS, GS, SOL, CL, SF)will not increment their pipeline statistics counters.							
29:24		<b>Reserved</b>							
		Access: RO							
		Format: MBZ							
23:16		<b>Render List Index</b>							
		Format: U8							
		Specifies the index in to the Render-List for the current Tile. Range [0..127]. HW will fetch the starting page offset for the visibility data of the current tile from below memory location [ {render_list_base_address[47:12], 12b0} + {render_list_pointer[31:6], 6b0} + (Render List Index <<1)]							
		<table border="1"> <thead> <tr> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>Render List Index must be set to 0 when 3DSTATE_TILE_PASS_INFO:Tile Count is 0x0 for the corresponding Tile Pass.</td> </tr> </tbody> </table>	Programming Notes	Render List Index must be set to 0 when 3DSTATE_TILE_PASS_INFO:Tile Count is 0x0 for the corresponding Tile Pass.					
Programming Notes									
Render List Index must be set to 0 when 3DSTATE_TILE_PASS_INFO:Tile Count is 0x0 for the corresponding Tile Pass.									

<b>3DSTATE_PTBR_TILE_SELECT_BODY</b>				
15:10	<b>Reserved</b>			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO			
Format:	MBZ			
9:0	<b>Tile Rect Array Index</b>			
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U10</td> </tr> </table>	Format:	U10	
	Format:	U10		
	<p>Specifies the index in to the Tile Rect Array of the current Tile Pass. Rang [0..1023].HW will fetch the RECT_STATE of the current tile from below memory location  [ {dynamic_state_base_addres[47:12], 12b0} + {Tile Rect Array Pointer[31:6], 6b0} + (Tile Index«3) ]</p>			
<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td>Tile Rect Array Index must be set to 0 when 3DSTATE_TILE_PASS_INFO:Tile Count is 0x0 for the corresponding Tile Pass.</td> </tr> </table>	<b>Programming Notes</b>	Tile Rect Array Index must be set to 0 when 3DSTATE_TILE_PASS_INFO:Tile Count is 0x0 for the corresponding Tile Pass.		
<b>Programming Notes</b>				
Tile Rect Array Index must be set to 0 when 3DSTATE_TILE_PASS_INFO:Tile Count is 0x0 for the corresponding Tile Pass.				

## 3DSTATE\_RASTER\_BODY

3DSTATE_RASTER_BODY			
Source:	RenderCS		
Size (in bits):	128		
Default Value:	0x00210000, 0x00000000, 0x00000000, 0x00000000		
DWord	Bit	Description	
0	31:29	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	28:27	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	26	<b>Viewport Z Far Clip Test Enable</b>	
	Format:	Enable	
	This field is used to control whether the Viewport Z Far extent is considered in VertexClipTest.		
	25	<b>Reserved</b>	
Access:		RO	
Format:		MBZ	
24	<b>Conservative Rasterization Enable</b>		
	Format:	Enable	
	This field when set enables conservative rasterization rules for all primitives except rectangles, points and lines. For rectangle, points and lines, setting this bit is ignored by hardware.		
	<b>Programming Notes</b>		
	This bit must not be set for primitives with poly-stippling enabled or native rectlist. When this bit is set, sampling mode must be set to "Centre" sampling i.e 3DSTATE_MULTISAMPLE::Pixel Location set to CENTER		
23:22	<b>API Mode</b>		
	Software sets this field according to the API's version. These bits are set for DX9 or OGL/DX10.0/DX10.1+/DX11.1 per the following values.		
	<b>Value</b>	<b>Name</b> <b>Description</b>	
	0h	DX9/OGL	
	1h	DX10.0	
	2h	DX10.1+	This is used for DX10.1+ and Vulkan API
3h	Reserved		
21	<b>Front Winding</b>		
	Determines whether a triangle object is considered "front facing" if the screen space vertex positions, when traversed in the order, result in a clockwise (CW) or counter-clockwise (CCW)		

## 3DSTATE\_RASTER\_BODY

		winding order. Does not apply to points or lines.	
		<b>Value</b>	<b>Name</b>
		0h	Clockwise
		1h	Counter Clockwise <b>[Default]</b>
			<b>Description</b>
			FRONTWINDING_CW
			FRONTWINDING_CCW
20:18	<b>Forced Sample Count</b>		
	Format:	U3	
	This field specifies how many samples/pixel exist for RT Independent Rasterization		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	NUMRASTSAMPLES_0	No RT Independent Rasterization
	1h	NUMRASTSAMPLES_1	1 rast-sample/pixel
	2h	NUMRASTSAMPLES_2	2 rast-samples/pixel
	3h	NUMRASTSAMPLES_4	4 rast-samples/pixel
	4h	NUMRASTSAMPLES_8	8 rast-samples/pixel
	5h	NUMRASTSAMPLES_16	16 rast-samples/pixel
	6h-7h	Reserved	
	<b>Programming Notes</b>		
	When 3DSTATE_MULTISAMPLE::Number of Multisamples != NUMSAMPLES_1, this field must be either NUMRASTSAMPLES_0 or NUMRASTSAMPLES_1.		
	When 3DSTATE_MULTISAMPLE::Number of Multisamples == NUMSAMPLES_1, this field must not be NUMRASTSAMPLES_1.		
17:16	<b>Cull Mode</b>		
	Controls removal (culling) of triangle objects based on orientation. The cull mode only applies to triangle objects and does not apply to lines, points or rectangles.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	CULLMODE_BOTH	All triangles are discarded (i.e., no triangle objects are drawn)
	1h	CULLMODE_NONE <b>[Default]</b>	No triangles are discarded due to orientation
	2h	CULLMODE_FRONT	Triangles with a front-facing orientation are discarded
	3h	CULLMODE_BACK	Triangles with a back-facing orientation are discarded
	<b>Programming Notes</b>		
	Orientation determination is based on the setting of the Front Winding state.		
15	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	

## 3DSTATE\_RASTER\_BODY

14	<b>Force Multisampling</b>	<p>This field provides a work around override for the computation of SF_INT::Multisample Rasterization Mode and WM_INT::Multisample Rasterization Mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Normal</td> <td>Multisampling mode is computed by HW according to formula for signal SF_INT::Multisample Rasterization Mode and WM_INT::Multisample Rasterization Mode in vol2a.11 3D Pipeline Windower &gt; Windower Pipelined State &gt; 3DSTATE_WM &gt; 3DSTATE_WM.</td> </tr> <tr> <td>1h</td> <td>Force</td> <td>Forces the DX Multisampling mode to be used directly</td> </tr> </tbody> </table>	Value	Name	Description	0h	Normal	Multisampling mode is computed by HW according to formula for signal SF_INT::Multisample Rasterization Mode and WM_INT::Multisample Rasterization Mode in vol2a.11 3D Pipeline Windower > Windower Pipelined State > 3DSTATE_WM > 3DSTATE_WM.	1h	Force	Forces the DX Multisampling mode to be used directly					
Value	Name	Description														
0h	Normal	Multisampling mode is computed by HW according to formula for signal SF_INT::Multisample Rasterization Mode and WM_INT::Multisample Rasterization Mode in vol2a.11 3D Pipeline Windower > Windower Pipelined State > 3DSTATE_WM > 3DSTATE_WM.														
1h	Force	Forces the DX Multisampling mode to be used directly														
13	<b>Smooth Point Enable</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>Software sets this according to API. When OGL and smooth point rasterization is required, this bit must be set. HW ignores this bit for primitives other than points.</p>	Format:	Enable												
Format:	Enable															
12	<b>DX Multisample Rasterization Enable</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>Software sets this according to the API's multisample enable</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; color: blue;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>This state only effects how the SF_INT/WM_INT::Multisample Rasterization Mode are set depending on some other states. This state mainly modifies the how the line rendering is done by setting SF_INT/WM_INT::Multisample Rasterization Mode to either OFF* or ON* . Please refer to table under SF_INT::Multisample Rasterization Mode.</td> </tr> </tbody> </table>	Format:	Enable	Programming Notes	This state only effects how the SF_INT/WM_INT::Multisample Rasterization Mode are set depending on some other states. This state mainly modifies the how the line rendering is done by setting SF_INT/WM_INT::Multisample Rasterization Mode to either OFF* or ON* . Please refer to table under SF_INT::Multisample Rasterization Mode.										
Format:	Enable															
Programming Notes																
This state only effects how the SF_INT/WM_INT::Multisample Rasterization Mode are set depending on some other states. This state mainly modifies the how the line rendering is done by setting SF_INT/WM_INT::Multisample Rasterization Mode to either OFF* or ON* . Please refer to table under SF_INT::Multisample Rasterization Mode.																
11:10	<b>DX Multisample Rasterization Mode</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>This field determines whether multisample rasterization is turned on/off, and how the pixel sample point(s) are defined. Software sets this according to the API's multisample state setting (if any)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>MSRASTMODE_OFF_PIXEL</td> </tr> <tr> <td>1h</td> <td>MSRASTMODE_OFF_PATTERN</td> </tr> <tr> <td>2h</td> <td>MSRASTMODE_ON_PIXEL</td> </tr> <tr> <td>3h</td> <td>MSRASTMODE_ON_PATTERN</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; color: blue;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>This field is used to directly set the SF_INT/WM_INT::Multisample Rasterization Mode when DX Multisample Rasterization Enable is set. Please refer to equation of SF_INT::Multisample Rasterization Mode.</td> </tr> </tbody> </table>	Format:	U2	Value	Name	0h	MSRASTMODE_OFF_PIXEL	1h	MSRASTMODE_OFF_PATTERN	2h	MSRASTMODE_ON_PIXEL	3h	MSRASTMODE_ON_PATTERN	Programming Notes	This field is used to directly set the SF_INT/WM_INT::Multisample Rasterization Mode when DX Multisample Rasterization Enable is set. Please refer to equation of SF_INT::Multisample Rasterization Mode.
Format:	U2															
Value	Name															
0h	MSRASTMODE_OFF_PIXEL															
1h	MSRASTMODE_OFF_PATTERN															
2h	MSRASTMODE_ON_PIXEL															
3h	MSRASTMODE_ON_PATTERN															
Programming Notes																
This field is used to directly set the SF_INT/WM_INT::Multisample Rasterization Mode when DX Multisample Rasterization Enable is set. Please refer to equation of SF_INT::Multisample Rasterization Mode.																
9	<b>Global Depth Offset Enable Solid</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>Enables computation and application of Global Depth Offset for SOLID objects.</p>	Format:	Enable												
Format:	Enable															

## 3DSTATE\_RASTER\_BODY

8	<b>Global Depth Offset Enable Wireframe</b>	
	Format:	Enable
	Enables computation and application of Global Depth Offset when triangles are rendered in WIREFRAME mode.	
7	<b>Global Depth Offset Enable Point</b>	
	Format:	Enable
	Enables computation and application of Global Depth Offset when triangles are rendered in POINT mode.	
6:5	<b>Front Face Fill Mode</b>	
	Format:	U2
	This state controls how front-facing triangle and rectangle objects are rendered.	
	<b>Value</b>	<b>Name</b> <b>Description</b>
	0h	SOLID      Any triangle or rectangle object found to be front-facing is rendered as a solid object. This setting is required when rendering rectangle (RECTLIST) objects.
	1h	WIREFRAME      Any triangle object found to be front-facing is rendered as a series of lines along the triangle boundaries (as determined by the topology type and controlled by the vertex EdgeFlags).
	2h	POINT      Any triangle object found to be front-facing is rendered as a set of point primitives at the triangle vertices (as determined by the topology type and controlled by the vertex EdgeFlags).
	3h	Reserved
4:3	<b>Back Face Fill Mode</b>	
	Format:	U2
	This state controls how back-facing triangle and rectangle objects are rendered.	
	<b>Value</b>	<b>Name</b> <b>Description</b>
	0h	SOLID      Any triangle or rectangle object found to be back-facing is rendered as a solid object. This setting is required when rendering rectangle (RECTLIST) objects.
	1h	WIREFRAME      Any triangle object found to be back-facing is rendered as a series of lines along the triangle boundaries (as determined by the topology type and controlled by the vertex EdgeFlags).
	2h	POINT      Any triangle object found to be back-facing is rendered as a set of point primitives at the triangle vertices (as determined by the topology type and controlled by the vertex EdgeFlags).
	3h	Reserved



<b>3DSTATE_RASTER_BODY</b>					
	2	<b>Antialiasing Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> <p>This field enables "alpha-based" line antialiasing.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>This field must be disabled if any of the render targets have integer (UINT or SINT) surface format.</p>	Format:	Enable	<b>Programming Notes</b>
	Format:	Enable			
	<b>Programming Notes</b>				
1	<b>Scissor Rectangle Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> <p>Enables operation of Scissor Rectangle.</p>	Format:	Enable		
Format:	Enable				
0	<b>Viewport Z Near Clip Test Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> <p>This field is used to control whether the Viewport Z Near extent is considered in VertexClipTest.</p>	Format:	Enable		
Format:	Enable				
1	31:0	<b>Global Depth Offset Constant</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">IEEE_FLOAT</td> </tr> </table> <p>Specifies the constant term in the Global Depth Offset function.</p>	Format:	IEEE_FLOAT	
Format:	IEEE_FLOAT				
2	31:0	<b>Global Depth Offset Scale</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">IEEE_FLOAT</td> </tr> </table> <p>Specifies the scale term used in the Global Depth Offset function.</p>	Format:	IEEE_FLOAT	
Format:	IEEE_FLOAT				
3	31:0	<b>Global Depth Offset Clamp</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">IEEE_FLOAT</td> </tr> </table> <p>Specifies the clamp term used in the Global Depth Offset function.</p>	Format:	IEEE_FLOAT	
Format:	IEEE_FLOAT				

## 3DSTATE\_SAMPLE\_MASK\_BODY

3DSTATE_SAMPLE_MASK_BODY												
Source:	RenderCS											
Size (in bits):	32											
Default Value:	0x00000000											
DWord	Bit	Description										
0	31:16	<b>Reserved</b>										
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
Access:	RO											
Format:	MBZ											
	15:0	<b>Sample Mask</b>										
		<table border="1"> <tr> <td>Format:</td> <td>Enable[16]</td> </tr> </table> <p>A per-multisample-position mask state variable that is immediately and unconditionally ANDed with the sample coverage mask as part of the rasterization process. This mask is applied prior to centroid selection. This mask must be ignored for centroid selection when RTIR is enabled i.e. <code>Forced_Sample_Count &gt; 0</code>.</p> <table border="1"> <thead> <tr> <th colspan="2">Programming Notes</th> </tr> </thead> <tbody> <tr> <td>•</td> <td>If <b>Number of Multisamples</b> is NUMSAMPLES_1, bits 15:1 of this field will be zeroed by HW.</td> </tr> <tr> <td>•</td> <td>If <b>Number of Multisamples</b> is NUMSAMPLES_2, bits 15:2 of this field will be zeroed by HW.</td> </tr> <tr> <td>•</td> <td>If <b>Number of Multisamples</b> is NUMSAMPLES_4, bits 15:4 of this field will be zeroed by HW.</td> </tr> <tr> <td>•</td> <td>If <b>Number of Multisamples</b> is NUMSAMPLES_8, bits 15:8 of this field will be zeroed by HW.</td> </tr> </tbody> </table> <p>When pixel shader writes to UAV but does not have actual render target write (i.e. no RT is bound to pixel shader, even though, RT write message is sent for EOT), appropriate <code>SAMPLE_MASK</code> must be all set depending on Number of Multisamples.</p>	Format:	Enable[16]	Programming Notes		•	If <b>Number of Multisamples</b> is NUMSAMPLES_1, bits 15:1 of this field will be zeroed by HW.	•	If <b>Number of Multisamples</b> is NUMSAMPLES_2, bits 15:2 of this field will be zeroed by HW.	•	If <b>Number of Multisamples</b> is NUMSAMPLES_4, bits 15:4 of this field will be zeroed by HW.
Format:	Enable[16]											
Programming Notes												
•	If <b>Number of Multisamples</b> is NUMSAMPLES_1, bits 15:1 of this field will be zeroed by HW.											
•	If <b>Number of Multisamples</b> is NUMSAMPLES_2, bits 15:2 of this field will be zeroed by HW.											
•	If <b>Number of Multisamples</b> is NUMSAMPLES_4, bits 15:4 of this field will be zeroed by HW.											
•	If <b>Number of Multisamples</b> is NUMSAMPLES_8, bits 15:8 of this field will be zeroed by HW.											

## 3DSTATE\_SAMPLER\_STATE\_POINTERS\_BODY

3DSTATE_SAMPLER_STATE_POINTERS_BODY					
Source:	RenderCS				
Size (in bits):	32				
Default Value:	0x00000000				
DWord	Bit	Description			
0	31:5	<p><b>Pointer to Sampler State</b></p> <table border="1"> <tr> <td>Format:</td> <td>DynamicStateOffset[31:5]SAMPLER_STATE*16</td> </tr> </table> <p>Specifies the 32-byte aligned address offset of the function's SAMPLER_STATE table. This offset is relative to the Dynamic State Base Address.</p>	Format:	DynamicStateOffset[31:5]SAMPLER_STATE*16	
	Format:	DynamicStateOffset[31:5]SAMPLER_STATE*16			
4:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				

## 3DSTATE\_SBE\_BODY

3DSTATE_SBE_BODY							
Source:	RenderCS						
Size (in bits):	160						
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000						
DWord	Bit	Description					
0	31	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO					
	Format:	MBZ					
	30	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO					
	Format:	MBZ					
	29	<b>Force Vertex URB Entry Read Length</b> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This field provides a work around override for the computation of SBE_INT::Vertex URB Entry Read Length. If asserted, 3DSTATE_SBE::Vertex URB Entry Read Length is be used directly. Otherwise, SBE_INT::Vertex URB Entry Read Length is computed normally.</p>	Format:	Enable			
	Format:	Enable					
	28	<b>Force Vertex URB Entry Read Offset</b> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This field provides a work around override for the computation of SBE_INT::Vertex URB Entry Read Offset. If asserted, 3DSTATE_SBE::Vertex URB Entry Read Offset is be used directly. Otherwise, SBE_INT::Vertex URB Entry Read Offset is computed normally.</p>	Format:	Enable			
	Format:	Enable					
27:22	<b>Number of SF Output Attributes</b> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Specifies the number of vertex attributes passed from the SF stage to the WM stage (does not include Position).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,32]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,32]	
Format:	U6						
Value	Name						
[0,32]							
21	<b>Attribute Swizzle Enable</b> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Enables the SF to perform swizzling on (up to the first 16) vertex attributes. If DISABLED, all vertex attributes are passed through.</p>	Format:	Enable				
Format:	Enable						
20	<b>Point Sprite Texture Coordinate Origin</b> <p>This state controls how Point Sprite Texture Coordinates are generated (when enabled on a per-attribute basis by Point Sprite Texture Coordinate Enable).</p>						

### 3DSTATE\_SBE\_BODY

		Value	Name	Description
		0h	UPPERLEFT	Top Left = (0,0,0,1)Bottom Left = (0,1,0,1)Bottom Right = (1,1,0,1)
		1h	LOWERLEFT	Top Left = (0,1,0,1)Bottom Left = (0,0,0,1)Bottom Right = (1,0,0,1)
19	<b>Primitive ID Override Component W</b>			
		Format:		Enable
	If set, the W component of output attribute selected by Primitive ID Override Attribute Select is overridden with the Primitive ID.			
18	<b>Primitive ID Override Component Z</b>			
		Format:		Enable
	If set, the Z component of output attribute selected by Primitive ID Override Attribute Select is overridden with the Primitive ID.			
17	<b>Primitive ID Override Component Y</b>			
		Format:		Enable
	If set, the Y component of output attribute selected by Primitive ID Override Attribute Select is overridden with the Primitive ID.			
16	<b>Primitive ID Override Component X</b>			
		Format:		Enable
	If set, the X component of output attribute selected by Primitive ID Override Attribute Select is overridden with the Primitive ID.			
15:11	<b>Vertex URB Entry Read Length</b>			
		Format:		U5
	Specifies the amount of URB data read for each Vertex URB entry, in 256-bit register increments.			
		Value	Name	
		[1,16]		
	<b>Programming Notes</b>			
	It is UNDEFINED to set this field to 0 indicating no Vertex URB data to be read. This field should be set to the minimum length required to read the maximum source attribute. The maximum source attribute is indicated by the maximum value of the enabled Attribute # Source Attribute if Attribute Swizzle Enable is set, Number of Output Attributes-1 if enable is not set. $read\_length = \text{ceiling}((\text{max\_source\_attr}+1)/2)$			
10:5	<b>Vertex URB Entry Read Offset</b>			
	Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB.			
4:0	<b>Primitive ID Override Attribute Select</b>			
	Specifies which attribute is overridden w/ the Primitive ID			

<b>3DSTATE_SBE_BODY</b>				
		<b>Programming Notes</b>		
		Set all Primitive ID Override Component Select X/Y/Z/W to 0 to indicate there is no Primitive ID override.		
1	31:0	<p><b>Point Sprite Texture Coordinate Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable[32]</td> </tr> </table> <p>When processing point primitives, the attributes from the incoming point vertex are typically copied to the point object corner vertices. However, if a bit is set in this field, the corresponding Attribute is selected as a Point Sprite Texture Coordinate, in which case each corner vertex is assigned a pre-defined texture coordinate as defined by the Point Sprite Texture Coordinate Origin state bit. Bit 0 corresponds to output Attribute 0.</p>	Format:	Enable[32]
Format:	Enable[32]			
2	31:0	<p><b>Constant Interpolation Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable[32]</td> </tr> </table> <p>This field is a bitmask containing a Constant Interpolation Enable bit for each corresponding attribute. If a bit is set, that attribute will undergo constant interpolation, and the corresponding WrapShortest Enable bits (if defined) will be ignored. If a bit is clear, components which are not enabled for WrapShortest interpolation (if defined) will be linearly interpolated.</p>	Format:	Enable[32]
Format:	Enable[32]			
3	31:30	<p><b>Attribute 15 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 15 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
	Format:	<b>Attribute_Component_Format</b>		
	29:28	<p><b>Attribute 14 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 14 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
	Format:	<b>Attribute_Component_Format</b>		
27:26	<p><b>Attribute 13 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 13 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>	
Format:	<b>Attribute_Component_Format</b>			
25:24	<p><b>Attribute 12 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 12 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>	
Format:	<b>Attribute_Component_Format</b>			

## 3DSTATE\_SBE\_BODY

23:22	<p><b>Attribute 11 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 11 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
21:20	<p><b>Attribute 10 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 10 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
19:18	<p><b>Attribute 9 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 9 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
17:16	<p><b>Attribute 8 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 8 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
15:14	<p><b>Attribute 7 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 7 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
13:12	<p><b>Attribute 6 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 6 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
11:10	<p><b>Attribute 5 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 5 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		

## 3DSTATE\_SBE\_BODY

	9:8	<b>Attribute 4 Active Component Format</b>	Format: <b>Attribute_Component_Format</b>	This state indicates which components of Attribute 4 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	7:6	<b>Attribute 3 Active Component Format</b>	Format: <b>Attribute_Component_Format</b>	This state indicates which components of Attribute 3 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	5:4	<b>Attribute 2 Active Component Format</b>	Format: <b>Attribute_Component_Format</b>	This state indicates which components of Attribute 2 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	3:2	<b>Attribute 1 Active Component Format</b>	Format: <b>Attribute_Component_Format</b>	This state indicates which components of Attribute 1 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	1:0	<b>Attribute 0 Active Component Format</b>	Format: <b>Attribute_Component_Format</b>	This state indicates which components of Attribute 0 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
4	31:30	<b>Attribute 31 Active Component Format</b>	Format: <b>Attribute_Component_Format</b>	This state indicates which components of Attribute 31 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.
	29:28	<b>Attribute 30 Active Component Format</b>	Format: <b>Attribute_Component_Format</b>	This state indicates which components of Attribute 30 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.



## 3DSTATE\_SBE\_BODY

27:26	<p><b>Attribute 29 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 29 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
25:24	<p><b>Attribute 28 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 28 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
23:22	<p><b>Attribute 27 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 27 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
21:20	<p><b>Attribute 26 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 26 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
19:18	<p><b>Attribute 25 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 25 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
17:16	<p><b>Attribute 24 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 24 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
15:14	<p><b>Attribute 23 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 23 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		

## 3DSTATE\_SBE\_BODY

13:12	<p><b>Attribute 22 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 22 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
11:10	<p><b>Attribute 21 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 21 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
9:8	<p><b>Attribute 20 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 20 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
7:6	<p><b>Attribute 19 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 19 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
5:4	<p><b>Attribute 18 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 18 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
3:2	<p><b>Attribute 17 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 17 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		
1:0	<p><b>Attribute 16 Active Component Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Format:</td> <td><b>Attribute_Component_Format</b></td> </tr> </table> <p>This state indicates which components of Attribute 16 are being used by the pixel shader Kernel. SBE will not perform attribute delta calculations for any disabled components. Operation is UNDEFINED if kernel uses attribute vertex delta for any disabled component.</p>	Format:	<b>Attribute_Component_Format</b>
Format:	<b>Attribute_Component_Format</b>		

## 3DSTATE\_SBE\_SWIZ\_BODY

3DSTATE_SBE_SWIZ_BODY		
Source:	RenderCS	
Size (in bits):	320	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..7	255:0	<b>Attribute</b> Format: <b>SF_OUTPUT_ATTRIBUTE_DETAIL[16]</b>
8..9	63:60	<b>Attribute 15 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	59:56	<b>Attribute 14 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	55:52	<b>Attribute 13 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	51:48	<b>Attribute 12 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	47:44	<b>Attribute 11 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	43:40	<b>Attribute 10 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	39:36	<b>Attribute 09 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	35:32	<b>Attribute 08 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	31:28	<b>Attribute 07 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	27:24	<b>Attribute 06 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	23:20	<b>Attribute 05 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	19:16	<b>Attribute 04 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	15:12	<b>Attribute 03 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>

<b>3DSTATE_SBE_SWIZ_BODY</b>		
	11:8	<b>Attribute 02 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	7:4	<b>Attribute 01 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>
	3:0	<b>Attribute 00 Wrap Shortest Enables</b> Format: <b>WRAP_SHORTEST_ENABLE</b>

## 3DSTATE\_SCISSOR\_STATE\_POINTERS\_BODY

3DSTATE_SCISSOR_STATE_POINTERS_BODY					
Source:	RenderCS				
Size (in bits):	32				
Default Value:	0x00000000				
DWord	Bit	Description			
0	31:5	<p><b>Scissor Rect Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>DynamicStateOffset[31:5]SCISSOR_RECT*16</td> </tr> </table> <p>Specifies the 32-byte aligned address offset of the SCISSOR_RECT state. This offset is relative to the <b>Dynamic State Base Address</b>.</p>	Format:	DynamicStateOffset[31:5]SCISSOR_RECT*16	
	Format:	DynamicStateOffset[31:5]SCISSOR_RECT*16			
4:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				

## 3DSTATE\_SF\_BODY

3DSTATE_SF_BODY					
Source:	RenderCS				
Size (in bits):	96				
Default Value:	0x00000000, 0x00000000, 0x00000800				
DWord	Bit	Description			
0	31:30	<b>Reserved</b>			
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
29:12	<b>Line Width</b>				
	<table border="1"> <tr> <td>Format:</td> <td>U11.7</td> </tr> </table> <p>Range: [0.0, 2047.9921875]</p> <p>Controls width of line primitives. Setting a Line Width of 0.0 specifies the rasterization of the "thinnest" (one-pixel-wide), non-antialiased lines. Note that this effectively overrides the effect of AAEnable (though the AAEnable state variable is not modified).</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>Software must not program a value of 0.0 when running in MSRASTMODE_ON_xxx modes - zero-width lines are not available when multisampling rasterization is enabled.</p>	Format:	U11.7		
Format:	U11.7				
11	<b>Legacy Global Depth Bias Enable</b>				
	<table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Enables the SF to use the Global Depth Offset Constant state unmodified. If this bit is not set, the SF will scale the Global Depth Offset Constant as described in section Error! Reference source not found. of this document.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This bit should be set whenever non zero depth bias (Slope, Bias) values are used. Setting this bit may have some degradation of performance for some workloads.</p>	Format:	Enable		
Format:	Enable				
10	<b>Statistics Enable</b>				
	<table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, this FF unit will increment CL_PRIMITIVES_COUNT on behalf of the CLIP stage. If DISABLED, CL_PRIMITIVES_COUNT will be left unchanged.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This bit should be set whenever clipping is enabled and the Statistics Enable bit is set in CLIP_STATE. It should be cleared if clipping is disabled or Statistics Enable in CLIP_STATE is clear.</p>	Format:	Enable		
Format:	Enable				
9:2	<b>Reserved</b>				
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				

<b>3DSTATE_SF_BODY</b>																					
	1	<b>Viewport Transform Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit controls the Viewport Transform function.</p>	Format:	Enable																	
	Format:	Enable																			
0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																
Access:	RO																				
Format:	MBZ																				
1	31	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ															
	Access:	RO																			
	Format:	MBZ																			
	30:29	<b>Deref Block Size</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2</td> </tr> </table> <p>Set the EODB batch size</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: 45%;">Name</th> <th style="width: 45%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Default Mode. Block deref size 32 <b>[Default]</b></td> <td>Block deref size of 32</td> </tr> <tr> <td>1h</td> <td>Per Poly Deref Mode when VS/DS have minimal handles.</td> <td>Per Poly. Deref mode when VS/DS have minimal handles.</td> </tr> <tr> <td>2h</td> <td>Block deref size 8</td> <td>8 handles mode, block size.</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="text-align: center; padding: 5px;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"> Deref Block size depends on the last enabled shader and number of handles programmed for that shader  1) For GS last shader enabled cases, the deref block is always set to a per poly(within hardware) If the last enabled shader is VS or DS.  1) If DS is last enabled shader then if the number of DS handles is less than 324, need to set per poly deref.  2) If VS is last enabled shader then if the number of VS handles is less than 192, need to set per poly deref </td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	Default Mode. Block deref size 32 <b>[Default]</b>	Block deref size of 32	1h	Per Poly Deref Mode when VS/DS have minimal handles.	Per Poly. Deref mode when VS/DS have minimal handles.	2h	Block deref size 8	8 handles mode, block size.	3h	Reserved		Programming Notes	Deref Block size depends on the last enabled shader and number of handles programmed for that shader 1) For GS last shader enabled cases, the deref block is always set to a per poly(within hardware) If the last enabled shader is VS or DS. 1) If DS is last enabled shader then if the number of DS handles is less than 324, need to set per poly deref. 2) If VS is last enabled shader then if the number of VS handles is less than 192, need to set per poly deref
	Format:	U2																			
	Value	Name	Description																		
	0h	Default Mode. Block deref size 32 <b>[Default]</b>	Block deref size of 32																		
	1h	Per Poly Deref Mode when VS/DS have minimal handles.	Per Poly. Deref mode when VS/DS have minimal handles.																		
	2h	Block deref size 8	8 handles mode, block size.																		
	3h	Reserved																			
Programming Notes																					
Deref Block size depends on the last enabled shader and number of handles programmed for that shader 1) For GS last shader enabled cases, the deref block is always set to a per poly(within hardware) If the last enabled shader is VS or DS. 1) If DS is last enabled shader then if the number of DS handles is less than 324, need to set per poly deref. 2) If VS is last enabled shader then if the number of VS handles is less than 192, need to set per poly deref																					
28	<b>Reserved</b>																				
27:18	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																
Access:	RO																				
Format:	MBZ																				
17:16	<b>Line End Cap Antialiasing Region Width</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2</td> </tr> </table> <p>This field specifies the distances over which the coverage of anti-aliased line end caps are computed.</p>	Format:	U2																		
Format:	U2																				

3DSTATE_SF_BODY		
Value	Name	Description
0h	0.5 pixels	0.5 pixels
1h	1.0 pixels	1.0 pixels
2h	2.0 pixels	2.0 pixels
3h	4.0 pixels	4.0 pixels
15:14	<b>Reserved</b>	
Access:	RO	
Format:	MBZ	
13	<b>Reserved</b>	
12	<b>Reserved</b>	
11:0	<b>Reserved</b>	
Access:	RO	
Format:	MBZ	
2	31	<b>Last Pixel Enable</b>
Format:	Enable	
If ENABLED, the last pixel of a diamond line will be lit. This state will only affect the rasterization of Diamond lines (will not affect wide lines or anti-aliased lines).		
<b>Programming Notes</b>		
Last pixel is applied to all lines of a LINELIST, and only the last line of a LINESTRIP.		
30:29	<b>Triangle Strip/List Provoking Vertex Select</b>	
Format:	U2	
Selects which vertex of a triangle (in a triangle strip or list primitive) is considered the "provoking vertex". Used for flat shading of primitives. Does current implementation send provoking vertex first?		
Value	Name	
0h	0	
1h	1	
2h	2	
3h	Reserved	
28:27	<b>Line Strip/List Provoking Vertex Select</b>	
Format:	U2	
Selects which vertex of a line (in a line strip or list primitive) is considered the "provoking vertex".		
Value	Name	Description
0h	0	Vertex 0
1h	1	Vertex 1
2h	Reserved	Reserved
3h	Reserved	Reserved



## 3DSTATE\_SF\_BODY

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

<b>3DSTATE_SF_BODY</b>		
	1h	State <b>[Default]</b> Use Point Width from State
10:0	<b>Point Width</b>	
	Format:	U8.3
	Range: [0.125, 255.875] pixels	
	This field specifies the size (width) of point primitives in pixels. This field is overridden (though not overwritten) whenever point width information is passed in the FVF	

## 3DSTATE\_SO\_BUFFER\_INDEX\_BODY

3DSTATE_SO_BUFFER_INDEX_BODY						
Source:	RenderCS					
Size (in bits):	224					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000					
DWord	Bit	Description				
0	31	<b>SO Buffer Enable</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>Enable</td></tr></table> If set, stream output to SO Buffer is enabled, , if 3DSTATE_STREAMOUT::SO Function ENABLE is also enabled..If clear, the SO Buffer is considered "not bound" and effectively treated as a zero-length buffer for the purposes of SO output and overflow detection. If an enabled stream's Stream to Buffer Selects includes this buffer it is by definition an overflow condition. That stream will cause no writes to occur, and only SO_PRIM_STORAGE_NEEDED[<stream>] will increment.		Enable		
		Enable				
	30:29	<b>Reserved</b> Access: <table border="1" style="display: inline-table;"><tr><td> </td><td>RO</td></tr></table> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>MBZ</td></tr></table>		RO		MBZ
		RO				
		MBZ				
	28:22	<b>SO Buffer Object Control State</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td><b>MEMORY_OBJECT_CONTROL_STATE</b></td></tr></table> Specifies the memory object control state for the SO buffer.		<b>MEMORY_OBJECT_CONTROL_STATE</b>		
		<b>MEMORY_OBJECT_CONTROL_STATE</b>				
	21	<b>Stream Offset Write Enable</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>Enable</td></tr></table> When set, this field allows the hardware to write SO_WRITE_OFFSET[n] as specified in the Stream Offset field. <table border="1" style="width: 100%; text-align: center;"><tr><td><b>Programming Notes</b></td></tr><tr><td>The field is operates irrespective of whether SO Buffer Enable is set or clear.</td></tr></table>		Enable	<b>Programming Notes</b>	The field is operates irrespective of whether SO Buffer Enable is set or clear.
		Enable				
	<b>Programming Notes</b>					
The field is operates irrespective of whether SO Buffer Enable is set or clear.						
20	<b>Stream Output Buffer Offset Address Enable</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>Enable</td></tr></table> When set, this field allows the hardware to read/write the stream output buffer offset as specified in the "Stream Output Buffer Offset Address" field. <table border="1" style="width: 100%; text-align: center;"><tr><td><b>Programming Notes</b></td></tr><tr><td>The field is operates irrespective of whether SO Buffer Enable is set or clear.</td></tr></table>		Enable	<b>Programming Notes</b>	The field is operates irrespective of whether SO Buffer Enable is set or clear.	
	Enable					
<b>Programming Notes</b>						
The field is operates irrespective of whether SO Buffer Enable is set or clear.						
19:0	<b>Reserved</b> Access: <table border="1" style="display: inline-table;"><tr><td> </td><td>RO</td></tr></table> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>MBZ</td></tr></table>		RO		MBZ	
	RO					
	MBZ					

<b>3DSTATE_SO_BUFFER_INDEX_BODY</b>			
1..2	63:2	<b>Surface Base Address</b>	
		Format: VIRTUAL_ADDR[63:2]	
		<b>Description</b>	
	VIRTUAL_ADDR[63:48] are ignored by the HW This field specifies the starting address of the buffer in Graphics Memory.		
1:0	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
3	31:30	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	29:0	<b>Surface Size</b>	
Format: U30-1 This field specifies the size of buffer in number DWords minus 1 of the buffer in Graphics Memory.			
4..5	63:2	<b>Stream Output Buffer Offset Address</b>	
		Format: VIRTUAL_ADDR[63:2]	
		<b>Description</b>	
	VIRTUAL_ADDR[63:48] are ignored by the HW This field specifies the starting address of the buffer in Graphics Memory where the Stream Output Buffer Offset is stored when all the data has been written. It is also used to fetch the stream Output buffer Offset when needed.		
1:0	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
6	31:0	<b>Stream Offset</b>	
This field specifies the Offset in stream output buffer to start at, or whether to append to the end of an existing buffer. The Offset must be DWORD aligned. If Stream Offset is equal to 0xFFFFFFFF then load the value at the Stream Output Buffer Offset address into SO_WRITE_OFFSET[Buffer#]. Otherwise, SO_WRITE_OFFSET[n] = Stream Offset.			

## 3DSTATE\_STENCIL\_BUFFER\_BODY

3DSTATE_STENCIL_BUFFER_BODY																							
Source:	RenderCS																						
Size (in bits):	224																						
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000																						
DWord	Bit	Description																					
0	31:29	<b>Surface Type</b>																					
		<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></td> </tr> <tr> <td>1h</td> <td>SURFTYPE_2D</td> <td>Defines a 2-dimensional map or array of maps</td> </tr> <tr> <td>2h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>3h</td> <td>SURFTYPE_CUBE</td> <td>Defines a cube map</td> </tr> <tr> <td>4h-6h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>7h</td> <td>SURFTYPE_NULL</td> <td>Defines a null surface</td> </tr> </tbody> </table>	Value	Name	Description	0h	Reserved		1h	SURFTYPE_2D	Defines a 2-dimensional map or array of maps	2h	Reserved		3h	SURFTYPE_CUBE	Defines a cube map	4h-6h	Reserved		7h	SURFTYPE_NULL	Defines a null surface
		Value	Name	Description																			
		0h	Reserved																				
		1h	SURFTYPE_2D	Defines a 2-dimensional map or array of maps																			
		2h	Reserved																				
		3h	SURFTYPE_CUBE	Defines a cube map																			
		4h-6h	Reserved																				
		7h	SURFTYPE_NULL	Defines a null surface																			
		<b>Programming Notes</b>																					
<p>The Surface Type of the Stencil buffer must be the same as the Surface Type of the 1. Render target(s) (defined in SURFACE_STATE), unless either the stencil buffer or render targets are SURFTYPE_NULL</p> <p>2. Depth buffer (defined in 3DSTATE_DEPTH_BUFFER) unless either the depth buffer or Stencil buffer surf_type are SURFTYPE_NULL.</p> <p>If stencil is enabled with 1D render target, stencil surface type needs to be set to 2D surface type and height set to 1. For this case only, the Surface Type of the stencil buffer can be 2D while the Surface Type of the render target(s) are 1D, representing an exception to a programming note above.</p>																							
<p><u>Issue</u> Semi pipelined flush not back pressuring when stencil buffer state is enabling thread dispatch.</p> <p><u>Workaround</u> <b>Astep only</b> An <b>additional pipe control with post-sync = store dword</b> operation would be required.( w/a is to have an <b>additional</b> pipe control after the stencil state whenever the surface state bits of this state is changing).</p>																							
28	<b>Stencil Write Enable</b>																						
	Format:	Enable																					
		This field enables stencil writes to the Stencil buffer surface. Both this field and the Stencil Buffer Write Enable field in DEPTH_STENCIL_STATE must be enabled in order for stencil writes to occur.																					
27	<b>Null Page Coherency Enable</b>																						
	Format:	Enable																					
		This field is used for enabling NULL coherency as defined under Tiled Resources.																					

<b>3DSTATE_STENCIL_BUFFER_BODY</b>		
	<b>Value</b>	<b>Name</b>
	1	Enable
	0	Disable <b>[Default]</b>
	<b>Programming Notes</b>	
	SW must enable this bit only if Tiled Resource is enabled	
26	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
25	<b>Stencil Compression Enable</b>	
	if enabled, indicates that Stencil Buffer Compression is Enabled When enabled, Stencil Buffer needs to be initialized via stencil clear (HZ_OP) before any renderpass.	
	<b>Programming Notes</b>	
	SW must set this bit if the Stencil Control surface enable is also set. The Stencil surface control enable is in Bit[24] of this DWORD.	
24	<b>Control Surface Enable</b>	
	If set to 1, it indicates if the common control surface is present. The read and write transaction opcodes sent by the clients (HZ, Z, STC) to the fabric are different depending on the control surface. If the control surface is not present, the reads and writes are in legacy mode. If the control surface is present, the reads and write opcodes will be either UNCOMPRESSED_TYP for uncompressible transactions or COMPRESSED_TYP for compressible transactions.	
	<b>Programming Notes</b>	
	SW must set this bit to "1", if the common control surface is present in the system.	
23	<b>Corner Texel Mode</b>	
	Format:	Enable
	This field, when ENABLED, indicates when a surface is using corner texel-mode for stencil surface. This bit changes how the size of each MIP when calculating the offset within a surface.	
	<b>Value</b>	<b>Name</b>
	0h	Disable <b>[Default]</b>
	1h	Enable
	<b>Description</b>	
	Corner Texel mode is not enabled.	
	Corner Texel Mode is enabled.	
	<b>Programming Notes</b>	
	Corner texel mode for the stencil buffer must be the same as the Corner texel mode of the 1. Render target(s) (defined in SURFACE_STATE), unless either the stencil buffer or render targets are SURFTYPE_NULL 2. Depth buffer (defined in 3DSTATE_DEPTH_BUFFER) unless either the depth buffer or Stencil buffer surf_type are SURFTYPE_NULL	

<b>3DSTATE_STENCIL_BUFFER_BODY</b>										
	22:18	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
	Access:	RO								
	Format:	MBZ								
17	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO									
Format:	MBZ									
16:0	<b>Surface Pitch</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U17-1</td> </tr> </table> <p>For TileYF and TileYS surfaces, the range is dependent on the Cu parameter (refer to <i>Memory Data Formats</i> section for the definition of the Cu parameter depending on the case). The range in bytes is <math>[2^{Cu}-1, 262143]</math> -&gt; <math>[(2^{Cu})B, 256KB] = [1 \text{ tile}, 256KB/(2^{Cu}) \text{ tiles}]</math></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>[7Fh, 1FFFFh]</td> <td></td> <td>corresponding to [128B, 256KB] also restricted to a multiple of 128B</td> </tr> </tbody> </table> <table border="1" style="width: 100%; text-align: center;"> <tr> <th style="background-color: #e1eef6;">Programming Notes</th> </tr> </table> <p>The pitch specified must be a multiple of the tile pitch, in the range [128B, 128KB].  The minimum pitch should be calculated as per the formula given below.  <i>The minimum pitch should be calculated based on Cu, Cv, W<sub>L</sub>.</i>  <i>The Cu, Cv are the tile constants and W<sub>L</sub> is the aligned width adjusted for MSAA.</i>  <i>use this for pitch formula :</i>  <math>Minimum\_pitch = (ceiling((W_0 * pixel\_size) / (1 \ll Cu)) * (1 \ll Cu)) / W_0</math> ; //W<sub>0</sub> is the aligned width for the largest LOD (i.e LOD 0)  <math>(1 \ll Cu) = \text{tile width in bytes}</math>  <math>(1 \ll Cv) = \text{tile height in lines}</math>  <i>Pixel_size = 1 (for STC buffer)</i></p>	Format:	U17-1	Value	Name	Description	[7Fh, 1FFFFh]		corresponding to [128B, 256KB] also restricted to a multiple of 128B	Programming Notes
Format:	U17-1									
Value	Name	Description								
[7Fh, 1FFFFh]		corresponding to [128B, 256KB] also restricted to a multiple of 128B								
Programming Notes										
1..2	63:0	<b>Surface Base Address</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>GraphicsAddress[63:0]</td> </tr> </table> <p>This field specifies address of the buffer in mapped Graphics Memory. Graphics Address [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%; text-align: center;"> <tr> <th style="background-color: #e1eef6;">Programming Notes</th> </tr> </table> <p>The stencil Buffer can only be mapped to Main Memory (uncached). If the surface is tiled, the base address must conform to the Per-Surface Tiling Alignment. If the buffer is linear, the surface must be 64-byte aligned.</p> <p>If the buffer is linear, the surface must be 64-byte aligned.</p>	Format:	GraphicsAddress[63:0]	Programming Notes					
Format:	GraphicsAddress[63:0]									
Programming Notes										
3	31	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO									
Format:	MBZ									

## 3DSTATE\_STENCIL\_BUFFER\_BODY

30:17	<b>Height</b>	Format: U14-1													
<p>This field specifies the height of the surface. If the surface is MIP-mapped, this field contains the height of the base MIP level.</p>															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 25%;">Description</th> <th style="width: 45%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Legal Range</td> <td>Height of surface - 1 (y/v dimension)</td> <td>(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_2D')</td> </tr> <tr> <td>[0,16383]</td> <td>Legal Range</td> <td>y/v dimension</td> <td>(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_CUBE')</td> </tr> </tbody> </table>				Value	Name	Description	Exists If	[0,16383]	Legal Range	Height of surface - 1 (y/v dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_2D')	[0,16383]	Legal Range	y/v dimension	(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_CUBE')
Value	Name	Description	Exists If												
[0,16383]	Legal Range	Height of surface - 1 (y/v dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_2D')												
[0,16383]	Legal Range	y/v dimension	(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_CUBE')												
<b>Programming Notes</b>															
<p>The Height of the stencil buffer must be the same as the</p> <ol style="list-style-type: none"> <li>Height of the render target(s) (defined in SURFACE_STATE), unless Surface Type is SURFTYPE_2D with Depth = 0 (non-array) and LOD = 0 (non-mip mapped).</li> <li>Depth buffer (defined in 3DSTATE_DEPTH_BUFFER) unless either the depth buffer or Stencil buffer surf_type are SURFTYPE_NULL</li> </ol>															
16	<b>Reserved</b>	Access: RO													
		Format: MBZ													
15	<b>Reserved</b>	Access: RO													
		Format: MBZ													
14:1	<b>Width</b>	Format: U14-1													
<p>This field specifies the width of the surface. If the surface is MIP-mapped, this field specifies the width of the base MIP level. The width is specified in units of pixels.</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: 25%;">Description</th> <th style="width: 45%;">Exists If</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td>Legal Range</td> <td>Width of surface - 1 (x/u dimension)</td> <td>(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_2D')</td> </tr> <tr> <td>[0,16383]</td> <td>Legal Range</td> <td>Width of surface - 1 (x/u dimension)</td> <td>(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_CUBE')</td> </tr> </tbody> </table>				Value	Name	Description	Exists If	[0,16383]	Legal Range	Width of surface - 1 (x/u dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_2D')	[0,16383]	Legal Range	Width of surface - 1 (x/u dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_CUBE')
Value	Name	Description	Exists If												
[0,16383]	Legal Range	Width of surface - 1 (x/u dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_2D')												
[0,16383]	Legal Range	Width of surface - 1 (x/u dimension)	(Structure[RENDER_SURFACE_STATE][Surface Type]== 'SURFTYPE_CUBE')												
<b>Programming Notes</b>															
<p>The Width specified by this field must be less than or equal to the surface pitch (specified in bytes via the Surface Pitch field). For cube maps, Width must be set equal to Height.</p> <ol style="list-style-type: none"> <li>The Width of the stencil buffer must be the same as the Width of the render target(s) (defined in SURFACE_STATE), unless Surface Type is SURFTYPE_1D or SURFTYPE_2D with Depth = 0 (non-array) and LOD = 0 (non-mip mapped).</li> <li>Depth buffer (defined in 3DSTATE_DEPTH_BUFFER) unless either the depth buffer or Stencil buffer surf_type are SURFTYPE_NULL</li> </ol>															



3DSTATE_STENCIL_BUFFER_BODY			
	0	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
4	31	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	30:20	<b>Depth</b>	
		Format:	U11-1
		This field specifies the total number of levels for a volume texture or the number of array elements allowed to be accessed starting at the Minimum Array Element for arrayed surfaces. If the volume texture is MIP-mapped, this field specifies the depth of the base MIP level.	
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	<b>Exists If</b>
		[0,2047]	Legal Range
		Number of array elements - 1	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')
		[0,0]	Legal Range
		Must be zero	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_CUBE')
		<b>Programming Notes</b>	
		The Depth of the Stencil buffer must be the same as 1. The Depth of the render target(s) (defined in SURFACE_STATE). 2. Depth buffer (defined in 3DSTATE_DEPTH_BUFFER) unless Depth buffer surf_type is SURFTYPE_NULL	
	19	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	18:8	<b>Minimum Array Element</b>	
		Format:	U11
		For 2D Surfaces: This field indicates the minimum array element that can be accessed as part of this surface. The delivered array index is added to this field before being used to address the surface. For Other Surfaces This field is ignored	
		<b>Value</b>	<b>Name</b>
		<b>Exists If</b>	
		[0,2047]	SURFTYPE_2D
		(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_1D' Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')	
		<b>Programming Notes</b>	
		Minimum array element of the Stencil buffer must be the same as the Surface Type of the 1. Render target(s) (defined in SURFACE_STATE), unless either the Stencil buffer or render targets are SURFTYPE_NULL 2. Depth buffer (defined in 3DSTATE_DEPTH_BUFFER) unless either the depth buffer or Stencil	

<b>3DSTATE_STENCIL_BUFFER_BODY</b>																	
		buffer surf_type are SURFTYPE_NULL															
	7	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ											
Access:	RO																
Format:	MBZ																
	6:0	<b>Stencil Buffer Object Control State</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td><b>MEMORY_OBJECT_CONTROL_STATE</b></td> </tr> </table> <p>Specifies the memory object control state for the stencil buffer.</p>	Format:	<b>MEMORY_OBJECT_CONTROL_STATE</b>													
Format:	<b>MEMORY_OBJECT_CONTROL_STATE</b>																
5	31:30	<b>Tiled Mode</b> For stencil Buffer Surfaces: This field specifies the tiled mode. For other surfaces: This field is ignored. <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NONE</td> <td>No tiled modes (TileYf, TileYs). Use Standard TileY</td> </tr> <tr> <td>1h</td> <td>TILEYF</td> <td>4KB tiled mode</td> </tr> <tr> <td>2h</td> <td>TILEYS</td> <td>64KB tiled mode</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> If <b>Tile Mode</b> is not set to TILEMODE_YMAJOR, this field must be set to TRMODE_NONE.	Value	Name	Description	0h	NONE	No tiled modes (TileYf, TileYs). Use Standard TileY	1h	TILEYF	4KB tiled mode	2h	TILEYS	64KB tiled mode	3h	Reserved	
Value	Name	Description															
0h	NONE	No tiled modes (TileYf, TileYs). Use Standard TileY															
1h	TILEYF	4KB tiled mode															
2h	TILEYS	64KB tiled mode															
3h	Reserved																
	29:26	<b>Mip Tail Start LOD</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U4</td> </tr> </table> <p><b>For Sampling Engine, Render Target, and Typed Surfaces:</b> This field indicates which LOD is the first one in the MIP tail if <b>Tiled Mode</b> 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.</p> <p><b>For other surfaces:</b> This field is ignored.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This field must be zero if the <b>Surface Format</b> is MONO8.            This field is ignored if <b>Tiled Mode</b> is TRMODE_NONE unless <b>Surface Type</b> is SURFTYPE_1D.</p> <p>If Tiled Mode is not TRMODE_NONE, this field must be set to ensure that mips within the mip tail do not overlap given the storage algorithms given in the Memory Data Formats section. The following table indicates the maximum size of the mip that is set to be the Mip Tail Start LOD for various cases:</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 20%;">Tiling Mode</th> <th style="width: 20%;">Slot Size in Bytes</th> <th style="width: 60%;">8-bit Size</th> </tr> </thead> <tbody> <tr> <td><b>2D TileYs 1x</b></td> <td>32KB</td> <td>(128, 256)</td> </tr> <tr> <td><b>2D TileYf 1x</b></td> <td>2KB</td> <td>(32, 64)</td> </tr> </tbody> </table>	Format:	U4	Tiling Mode	Slot Size in Bytes	8-bit Size	<b>2D TileYs 1x</b>	32KB	(128, 256)	<b>2D TileYf 1x</b>	2KB	(32, 64)				
Format:	U4																
Tiling Mode	Slot Size in Bytes	8-bit Size															
<b>2D TileYs 1x</b>	32KB	(128, 256)															
<b>2D TileYf 1x</b>	2KB	(32, 64)															
	25:6	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ											
Access:	RO																
Format:	MBZ																

<b>3DSTATE_STENCIL_BUFFER_BODY</b>					
	5:0	<b>Reserved</b>			
		Access:	RO		
		Format:	MBZ		
6	31:21	<b>Render Target View Extent</b>			
		Format:	U11-1		
		<b>Value</b>	<b>Name</b>	<b>Description</b>	<b>Exists If</b>
		[0,2047]	Legal Range	Number of array elements- 1	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_2D')
		[0,0]	Legal Range	Must be zero	(Structure[RENDER_SURFACE_STATE][Surface Type]='SURFTYPE_CUBE')
		<b>Programming Notes</b>			
		Render Target View Ext of the stencil buffer must be the same as the Surface Type of the			
		1. Render target(s) (defined in SURFACE_STATE), unless either the depth buffer or render targets are SURFTYPE_NULL			
		2. Depth buffer (defined in 3DSTATE_DEPTH_BUFFER) unless either the depth buffer or Stencil buffer surf_type are SURFTYPE_NULL			
			20	<b>Reserved</b>	
Access:	RO				
		Format:	MBZ		
	19:16	<b>Surf LOD</b>			
		Format:	U4		
		LOD units			
		<b>Value</b>	<b>Name</b>		
		[0-14]			
		<b>Programming Notes</b>			
		Surf LOD of the stencil buffer must be the same as the Surface Type of the			
		1. Render target(s) (defined in SURFACE_STATE), unless either the stencil buffer or render targets are SURFTYPE_NULL			
		2. Depth buffer (defined in 3DSTATE_DEPTH_BUFFER) unless either the depth buffer or Stencil buffer surf_type are SURFTYPE_NULL			
			15	<b>Reserved</b>	
Access:	RO				
		Format:	MBZ		

## 3DSTATE\_STENCIL\_BUFFER\_BODY

14:0	<b>Surface QPitch</b>	
	Format:	U17[16:2]
	<p>The interpretation of this field is dependent on <b>Surface Type</b> as follows:</p> <ul style="list-style-type: none"> <li>SURFTYPE_2D/CUBE: distance in <i>rows</i> between array slices.</li> </ul>	
	Other surface types: field is ignored	
	Format:	QPitch[16:2]
	<b>Value</b>	<b>Name</b>
	<b>Description</b>	
	[1h,7FFFh]	in multiples of 4 (low 2 bits missing)
	<b>Programming Notes</b>	
	<p>For 2D Surfaces: This field must be set to an integer multiple of 8. Software must ensure that this field is set to a value sufficiently large that array slices in the surface do not overlap. Refer to the Memory Data Formats section for information on how surfaces are stored.</p> <p>TYS/TYF QPitch is valid only for 2D array surfaces and represents the tile-padded total number of texels(lines) in a single array slice.</p> <p>Height of each LOD:</p> <p><math>HL = \text{AlignToTileHeight}( \text{MSAA\_height\_factor} * (\mathbf{height}\gg L) &gt; 0 ? \mathbf{height}\gg L : 1)</math>, where <math>\text{AlignToTileHeight}(x)</math> is <math>(\text{ceiling}(x) / (1 \ll Cv)) * (1 \ll Cv)</math></p> <p>Height of all LODs is a sum:</p> <p><math>H = H_0 + H_1 + \dots + H_n</math>,</p> <p>N is number of mip levels.</p> <p>If surface has MIP tail, equation stops at <math>H_n</math> where <math>n = \text{MipTailStartLOD}</math>. MipTail is single tile.</p> <p>QPitch is multiple of tile height <math>(1 \ll Cv)</math> and should be equal or greater H computed above.</p>	

## 3DSTATE\_STREAMOUT\_BODY

3DSTATE_STREAMOUT_BODY								
Source:	RenderCS							
Size (in bits):	128							
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000							
DWord	Bit	Description						
0	31	<p><b>SO Function Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>If set, the SO function is enabled. Vertex data will be streamed out to memory (subject to overflow detection) as controlled by the various SO-related state variables. If clear, the SO function is disabled, and therefore no vertex data will be streamed out to memory. However, the Rendering Disable and Render Stream Select fields will still be used to determine which vertices (if any) are forwarded down the pipeline for (possible) rendering.</p>	Format:	U1				
	Format:	U1						
	30	<p><b>API Rendering Disable</b></p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>If set, Indicates the API wants the SO stage not to forward any topologies down the pipeline. If clear, Indicates the API wants the SO stage to forward topologies associated with <b>Render Stream Select</b> down the pipeline. This bit is used even if <b>SO Function Enable</b> is DISABLED.</p> <table border="1"> <tr> <th colspan="2">Programming Notes</th> </tr> <tr> <td colspan="2">The SOL unit generates an SOL_INT::Render_Enable which ultimately controls whether rendering occurs or not.</td> </tr> </table>	Format:	U1	Programming Notes		The SOL unit generates an SOL_INT::Render_Enable which ultimately controls whether rendering occurs or not.	
	Format:	U1						
	Programming Notes							
The SOL unit generates an SOL_INT::Render_Enable which ultimately controls whether rendering occurs or not.								
29	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO							
Format:	MBZ							
28:27	<p><b>Render Stream Select</b></p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field specifies which stream has been selected to be forwarded down the pipeline for possible rendering. Topologies from other streams will not be passed down the pipeline. If Rendering Disable is set, this field is ignored, as no topologies are sent down the pipeline.</p> <p>SO Function Enable must also be ENABLED in order for this field to select a stream for rendering. When <b>SO Function Enable</b> is DISABLED and Rendering Disable is cleared (i.e., rendering is enabled), StreamID is ignored downstream of the SO stage, allowing any stream to be rendered.</p>	Format:	U2					
Format:	U2							
26	<p><b>Reorder Mode</b></p> <p>This bit controls how vertices of triangle objects in TRISTRIP[_ADJ] and TRISTRIP_REV are reordered for the purposes of stream-out only (does not impact rendering). See table in Input Buffering.</p>							

## 3DSTATE\_STREAMOUT\_BODY

Value	Name	Description
0h	LEADING	Reorder the vertices of alternating triangles of a TRISTRIP[_ADJ] such that the leading (first) vertices are in consecutive order starting at v0. A similar reordering is performed on alternating triangles in a TRISTRIP_REV.
1h	TRAILING	Reorder the vertices of alternating triangles of a TRISTRIP[_ADJ] such that the trailing (last) vertices are in consecutive order starting at v2. A similar reordering is performed on alternating triangles in a TRISTRIP_REV.
<b>25 SO Statistics Enable</b>		
Format:		Enable
This bit controls whether StreamOutput statistics register(s) can be incremented.		
Value	Name	Description
0h	Disable	SO_NUM_PRIMS_WRITTEN[0..3] and SO_PRIM_STORAGE_NEEDED[0..3] registers cannot increment.
1h	Enable	SO_NUM_PRIMS_WRITTEN[0..3] and SO_PRIM_STORAGE_NEEDED[0..3] registers can increment.
<b>24:23 Force Rendering</b>		
This field provides a work around override for the computation of SOL_INT::Render_Enable		
Value	Name	Description
0h	Normal	SOL_INT::Render_Enable is computed normally
1h	Reserved	
2h	Force_Off	Forces the rendering to be disabled.
3h	Force_on	Forces the rendering to be enabled.
<b>22:0 Reserved</b>		
Access:		RO
Format:		MBZ
1	<b>31:30 Reserved</b>	
	Access: RO	
	Format: MBZ	
<b>29 Stream 3 Vertex Read Offset</b>		
Format:		U1
Specifies amount of data to skip over before reading back Stream 3 vertex data. (See <b>Stream 0 Vertex Read Offset</b> )		
<b>28:24 Stream 3 Vertex Read Length</b>		
Format:		U5-1
(See Stream 0 Vertex Read Length)		
<b>23:22 Reserved</b>		
Access:		RO
Format:		MBZ

<b>3DSTATE_STREAMOUT_BODY</b>	
21	<b>Stream 2 Vertex Read Offset</b> Format: U1 Specifies amount of data to skip over before reading back Stream 2 vertex data. (See Stream 0 Vertex Read Offset)
	<b>Stream 2 Vertex Read Length</b> Format: U5-1
	<b>Reserved</b> Access: RO Format: MBZ
	<b>Stream 1 Vertex Read Offset</b> Format: U1 Specifies amount of data to skip over before reading back Stream 1 vertex data. (See Stream 0 Vertex Read Offset)
	<b>Stream 1 Vertex Read Length</b> Format: U5-1 (See Stream 0 Vertex Read Length)
	<b>Reserved</b> Access: RO Format: MBZ
	<b>Stream 0 Vertex Read Offset</b> Format: U1 Specifies amount of data to skip over before reading back Stream 0 vertex data. Must be zero if the GS is enabled and the Output Vertex Size field in 3DSTATE_GS is programmed to 0 (i.e., one 16B unit).
	<b>Stream 0 Vertex Read Length</b> Format: U5-1 Specifies amount of vertex data to read back for Stream 0 vertices, starting at the Stream 0 Vertex Read Offset location. Maximum readback is 17 256-bit units (34 128-bit vertex attributes). Read data past the end of the valid vertex data has undefined contents, and therefore shouldn't be used to source stream out data. Must be zero (i.e., read length = 256b) if the GS is enabled and the Output Vertex Size field in 3DSTATE_GS is programmed to 0 (i.e., one 16B unit).
	<b>Reserved</b> Access: RO Format: MBZ
	27:16

<b>3DSTATE_STREAMOUT_BODY</b>						
	15:12	<b>Reserved</b>				
		Access: RO				
		Format: MBZ				
	11:0	<b>Buffer 0 Surface Pitch</b>				
		Format: U12				
		This field specifies the pitch of the SO buffer in #Bytes.				
		<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>[0,2048]</td> <td>Must be 0 or a multiple of 4 Bytes.</td> </tr> </tbody> </table>	Value	Name	[0,2048]	Must be 0 or a multiple of 4 Bytes.
	Value	Name				
	[0,2048]	Must be 0 or a multiple of 4 Bytes.				
		<b>Programming Notes</b>				
	A Surface Pitch of 0 indicates an un-bound buffer. No writes are performed. Surface Base Address is ignored.					
3	31:28	<b>Reserved</b>				
		Access: RO				
		Format: MBZ				
	27:16	<b>Buffer 3 Surface Pitch</b>				
		Format: U12				
	15:12	<b>Reserved</b>				
		Access: RO				
		Format: MBZ				
	11:0	<b>Buffer 2 Surface Pitch</b>				
		Format: U12				



## 3DSTATE\_TE\_BODY

3DSTATE_TE_BODY											
Source:		RenderCS									
Size (in bits):		128									
Default Value:		0x00000000, 0x00000000, 0x00000000, 0x00000000									
DWord	Bit	Description									
0	31:24	<b>Reserved</b>									
		Access:						RO			
		Format:						MBZ			
	23:2	<b>Patch Header Layout</b>									
		This field describes the layout of the tessellation factor DWORDS in the patch header. The layout depends on the value of this field and the TE Domain.									
Value		Name	Description							Programming Notes	
0h		LEGACY		DW7	DW6	DW5	DW4	DW3	DW2	DW1	DW0
		QUAD	UEQ0	UEQ1	VEQ0	VEQ1	Inside U	Inside V	-	-	
		TRI	UEQ0	VEQ0	WEQ0	Inside	-	-	-	-	
		ISOLINE	Line Detail	Line Density	-	-	-	-	-	-	
2h	REVERSED		DW7	DW6	DW5	DW4	DW3	DW2	DW1	DW0	
		QUAD	-	-	Inside V	Inside U	VEQ1	VEQ0	UEQ1	UEQ0	
		TRI	-	-	-	-	Inside	WEQ0	VEQ0	UEQ0	
		ISOLINE	-	-	-	-	-	-	Line Density	Line Detail	
3h	REVERSED_TRI_INSIDE_SEPARATE		DW7	DW6	DW5	DW4	DW3	DW2	DW1	DW0	This layout may only be used with a TE Domain of TRI.
		TRI	-	-	-	Inside	-	WEQ0	VEQ0	UEQ0	
21	<b>Reserved</b>										
	Access:						RO				
	Format:						MBZ				

## 3DSTATE\_TE\_BODY

20	<b>Tessellation Factor Format</b>		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	FLOAT32	The tessellation factors in the patch header are in a FLOAT32 format.
	1h	FLOAT16	The tessellation factors in the patch header are in a FLOAT16 format. The tessellation factors still occupy the same DWORD as with the FLOAT32 format, but occupy only bits [15:0] of the DWORD.
19	<b>Tessellation Scale Factor Enable</b>		
	Format:	Enable	
	If ENABLED, the tessellation factors will be multiplied by the Tessellation Scale Factor.		
	<b>Programming Notes</b>		
Note that if ENABLED, the Tessellation Factor Format must be FLOAT16.			
18:1 7	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
16	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
15:1 4	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
13:1 2	<b>Partitioning</b>		
	Format:	U2	
	This field specifies how edges are partitioned based on tessellation factor.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	INTEGER	Outside/inside edges are divided into an integer number of equal-sized segments.
1h	ODD_FRACTIONAL	Outside/inside edges are divided into an odd number of possibly-unequal-sized segments.	
2h	EVEN_FRACTIONAL	Outside/inside edges are divided into an even number of possibly-unequal-sized segments.	
3h	POW2	Outside/inside edges are divided into a power of 2 number of equal-sized segments.	
11:1 0	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	

## 3DSTATE\_TE\_BODY

9:8	<b>Output Topology</b>		
		Format:	U2
This field specifies which primitive types are to be output.			
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
0h	POINT	Points are output (as POINTLIST topologies)	
1h	LINE	Lines are output (as LINESTRIP topologies). Only valid if ISOLINE domain is selected.	
2h	TRI_CW	Clockwise-ordered triangles are output (either as TRISTRIP, TRISTRIP_REV or TRILIST topologies). Not valid if ISOLINE domain is selected.	
3h	TRI_CCW	Count-clockwise-ordered triangles are output (either as TRISTRIP, TRISTRIP_REV or TRILIST topologies). Not valid if ISOLINE domain is selected.	
7:6	<b>Reserved</b>		
		Access:	RO
		Format:	MBZ
5:4	<b>TE Domain</b>		
		Format:	U2
This field specifies which type of domain is to be tessellated.			
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
0h	QUAD	2D (U, V) domain is tessellated	
1h	TRI	Triangular (U, V, W) domain is tessellated	
2h	ISOLINE	2D (U, V) domain is tessellated.	
3	<b>Reserved</b>		
		Access:	RO
		Format:	MBZ
2:1	<b>TE Mode</b>		
		Format:	U2
When TE Enable is ENABLED, this field specifies the overall operation of the TE stage. This field is ignored if TE Enable is DISABLED.			
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
0h	HW_TESS	Normal HW Tessellation Mode. The TessFactors are read from the patch URB entry, and are used to perform fixed-function hardware tessellation of the specified domain.	

<b>3DSTATE_TE_BODY</b>												
0	<p><b>TE Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, the TE stage will perform tessellation processing on incoming patch primitives. The TE Mode field determines how this tessellation operation proceeds. If DISABLED, the TE goes into pass-through mode. All other state fields are ignored.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.</td> </tr> </table>	Format:	Enable	Programming Notes	The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.							
Format:	Enable											
Programming Notes												
The tessellation stages (HS, TE and DS) must be enabled/disabled as a group. I.e., draw commands can only be issued if all three stages are enabled or all three stages are disabled, otherwise the behavior is UNDEFINED.												
1	<p>31:0 <b>Maximum Tessellation Factor Odd</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>This field specifies the maximum TessFactor for ODD_FRACTIONAL partitioning when in HW_TESS mode.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <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;">[3F800000h,427C0000h]</td> <td style="text-align: center;">[1,63]</td> <td>Value can be set between [1,63]. Value must be a IEEE_Float representation of an odd integer.</td> </tr> </tbody> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>Note that ISOLINE's LineDensity TF is always subjected to INTEGER partitioning regardless of the Partitioning state.</td> </tr> </table>	Format:	IEEE_FLOAT	Value	Name	Description	[3F800000h,427C0000h]	[1,63]	Value can be set between [1,63]. Value must be a IEEE_Float representation of an odd integer.	Programming Notes	Note that ISOLINE's LineDensity TF is always subjected to INTEGER partitioning regardless of the Partitioning state.	
Format:	IEEE_FLOAT											
Value	Name	Description										
[3F800000h,427C0000h]	[1,63]	Value can be set between [1,63]. Value must be a IEEE_Float representation of an odd integer.										
Programming Notes												
Note that ISOLINE's LineDensity TF is always subjected to INTEGER partitioning regardless of the Partitioning state.												
2	<p>31:0 <b>Maximum Tessellation Factor Not Odd</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>This field specifies the maximum TessFactor for EVEN_FRACTIONAL, INTEGER or POW2 partitioning when in HW_TESS mode.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <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;">[40000000h,42800000h]</td> <td style="text-align: center;">[2,64]</td> <td>Value can be set between [2,64]. Value must be a IEEE_Float representation of an even integer.</td> </tr> </tbody> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>Note that ISOLINE's LineDensity TF is always subjected to INTEGER partitioning regardless of the Partitioning state.</td> </tr> <tr> <td>If Partitioning is set to POW2, this field must be programmed to a power of 2 number.</td> </tr> </table>	Format:	IEEE_FLOAT	Value	Name	Description	[40000000h,42800000h]	[2,64]	Value can be set between [2,64]. Value must be a IEEE_Float representation of an even integer.	Programming Notes	Note that ISOLINE's LineDensity TF is always subjected to INTEGER partitioning regardless of the Partitioning state.	If Partitioning is set to POW2, this field must be programmed to a power of 2 number.
Format:	IEEE_FLOAT											
Value	Name	Description										
[40000000h,42800000h]	[2,64]	Value can be set between [2,64]. Value must be a IEEE_Float representation of an even integer.										
Programming Notes												
Note that ISOLINE's LineDensity TF is always subjected to INTEGER partitioning regardless of the Partitioning state.												
If Partitioning is set to POW2, this field must be programmed to a power of 2 number.												
3	<p>31:0 <b>Tessellation Scale Factor</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>If Tessellation Scale Factor Enable is ENABLED, the tessellation factors in the patch header will be multiplied by this value.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>This FLOAT32 value will be converted to a FLOAT16 value by hardware prior to scaling.</td> </tr> <tr> <td>This FLOAT32 value must <b>NOT</b> be negative, zero, denormal, infinite, or a NaN.</td> </tr> </table>	Format:	IEEE_FLOAT	Programming Notes	This FLOAT32 value will be converted to a FLOAT16 value by hardware prior to scaling.	This FLOAT32 value must <b>NOT</b> be negative, zero, denormal, infinite, or a NaN.						
Format:	IEEE_FLOAT											
Programming Notes												
This FLOAT32 value will be converted to a FLOAT16 value by hardware prior to scaling.												
This FLOAT32 value must <b>NOT</b> be negative, zero, denormal, infinite, or a NaN.												

## 3DSTATE\_URB\_ALLOC\_DS\_BODY

DW ord		Bit	Description
Source:			RenderCS
Size (in bits):			64
Default Value:			0x00000000, 0x00000000
0	31:29	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	28:21	<b>DS URB Starting Address SliceN</b>	
		Format:	U8
<p>This field specifies the offset (from the start of the URB memory in additional slices) of the DS URB allocation, specified in multiples of 8 KB. For each additional, enabled slice, HW will increase the DS URB Starting Address by the value specified, resulting in an DS URB Starting Address within the total URB space.</p>			
		<b>Value</b>	<b>Name</b>
		[0,127]	
<b>Programming Notes</b>			
This field is ignored by HW as there is only one slice in the device.			
	20:18	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	17:10	<b>DS URB Starting Address Slice0</b>	
		Format:	U8
<p>This field specifies the offset (from the start of Slice0 URB memory) of the DS URB Handle allocation for Slice0, specified in multiples of 8 KB. This address must account for any Push Constant allocations, as those allocations begin at offset 0 in URB memory.</p>			
		<b>Value</b>	<b>Programming Notes</b>
		[0,127]	The DS URB Starting Address Slice 0 must be greater than the render and posh push constant space allocated using 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_GS, 3DSTATE_PUSH_CONSTANT_ALLOC_PS and VSR_PUSH_CONSTANT_BASE.
	9:0	<b>DS URB Entry Allocation Size</b>	
		Format:	U10-1
Specifies the size, count of 512-bit units, of each URB entry owned by DS.			

		<b>3DSTATE_URB_ALLOC_DS_BODY</b>	
		Value	Name
		[0,511]	
1	31:16	<b>DS Number of URB Entries SliceN</b>	
		Format:	U16
		This field specifies the number of URB entries in slices beyond Slice0 to be allocated to DS.	
		Value	Name
		[0,3576]	
		Programming Notes	
		This field is ignored by HW as there is only one slice in the device.	
		DS Number of URB Entries must be divisible by 8 if the DS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"	
	15:0	<b>DS Number of URB Entries Slice0</b>	
		Format:	U16
		This field specifies the number of URB entries in Slice0 URB memory to be allocated to DS.	
		Value	Name
		[0,3576]	
		Programming Notes	
		DS Number of URB Entries must be divisible by 8 if the DS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"	

## 3DSTATE\_URB\_ALLOC\_GS\_BODY

3DSTATE_URB_ALLOC_GS_BODY				
Source:	RenderCS			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:29	<b>Reserved</b>		
		Access: RO		
		Format: MBZ		
	28:21	<b>GS URB Starting Address SliceN</b>		
		Format: U8		
This field specifies the offset (from the start of the URB memory in slices beyond Slice0) of the GS URB allocation, specified in multiples of 8 KB.				
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,127]</td> <td></td> </tr> </tbody> </table>		Value	Name	[0,127]
Value	Name			
[0,127]				
<b>Programming Notes</b>				
This field is ignored by HW as there is only one slice in the device.				
20:18	<b>Reserved</b>			
	Access: RO			
	Format: MBZ			
17:10	<b>GS URB Starting Address Slice0</b>			
	Format: U8			
	This field specifies the offset (from the start of Slice0 URB memory) of the GS URB allocation, specified in multiples of 8 KB. This address must account for any Push Constant allocations, as those allocations begin at offset 0 in Slice0 URB memory.			
	<table border="1"> <thead> <tr> <th>Value</th> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>[0,127]</td> <td>The GS URB Starting Address Slice 0 must be greater than the render and posh push constant space allocated using 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_GS, 3DSTATE_PUSH_CONSTANT_ALLOC_PS and VSR_PUSH_CONSTANT_BASE.</td> </tr> </tbody> </table>	Value	Programming Notes	[0,127]
Value	Programming Notes			
[0,127]	The GS URB Starting Address Slice 0 must be greater than the render and posh push constant space allocated using 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_GS, 3DSTATE_PUSH_CONSTANT_ALLOC_PS and VSR_PUSH_CONSTANT_BASE.			
9:0	<b>GS URB Entry Allocation Size</b>			
	Format: U10-1			
	Specifies the size, count of 512-bit units, of each URB entry owned by GS in units of 64B.			
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,511]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,511]
Value	Name			
[0,511]				

<b>3DSTATE_URB_ALLOC_GS_BODY</b>								
1	31:16	<p><b>GS Number of URB Entries SliceN</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U16</td> </tr> </table> <p>This field specifies the number of URB entries in slices beyond Slice0 to be allocated to GS. SW shall ensure that the GS Number of URB Entries does not exceed the relevant ValidValue range listed below.</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>[0,1548]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This field is ignored by single-slice devices.</p> <p>GS Number of URB Entries must be divisible by 8 if the GS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"</p> <p>This field is ignored by HW as there is only one slice in the device.</p>	Format:	U16	Value	Name	[0,1548]	
	Format:	U16						
Value	Name							
[0,1548]								
15:0	<p><b>GS Number of URB Entries Slice0</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U16</td> </tr> </table> <p>This field specifies the number of URB entries in Slice0 URB memory to be allocated to GS. SW shall ensure that the GS Number of Entries does not exceed the relevant ValidValue range listed below.</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>[0,1548]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>GS Number of URB Entries must be divisible by 8 if the GS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"</p>	Format:	U16	Value	Name	[0,1548]		
Format:	U16							
Value	Name							
[0,1548]								



## 3DSTATE\_URB\_ALLOC\_HS\_BODY

3DSTATE_URB_ALLOC_HS_BODY			
Source:	RenderCS		
Size (in bits):	64		
Default Value:	0x00000000, 0x00000000		
DWord	Bit	Description	
0	31:29	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	28:21	HS URB Starting Address SliceN	Format: U8
			This field specifies the offset (from the start of the URB memory in slices beyond Slice0) of the HS URB allocation, specified in multiples of 8 KB.
<b>Value</b>		<b>Name</b>	
[0,127]			
<b>Programming Notes</b>		This field is ignored by HW as there is only one slice in the device.	
20:18	<b>Reserved</b>	Access: RO	
		Format: MBZ	
17:10	HS URB Starting Address Slice0	Format: U8	
		This field specifies the offset (from the start of Slice0 URB memory) of the HS URB allocation, specified in multiples of 8 KB. This address must account for any Push Constant allocations, as those allocations begin at offset 0 in Slice0 URB memory.	
	<b>Value</b>	<b>Programming Notes</b>	
	[0,127]	The VS URB Starting Address Slice 0 must be greater than the render and posh push constant space allocated using 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_GS, 3DSTATE_PUSH_CONSTANT_ALLOC_PS and VSR_PUSH_CONSTANT_BASE.	
9:0	HS URB Entry Allocation Size	Format: U10-1	
		Specifies the size, count of 512-bit units, of each URB entry owned by HS.	
	<b>Value</b>	<b>Name</b>	
	[0,511]		

<b>3DSTATE_URB_ALLOC_HS_BODY</b>								
1	31:16	<p><b>HS Number of URB Entries SliceN</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>This field specifies the number of URB entries in slices beyond Slice0 to be allocated to HS. SW shall ensure that the total HS Number of URB Entries does not exceed the relevant ValidValue range listed below.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 60%;">Value</th> <th style="width: 40%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,1548]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This field is ignored by HW as there is only one slice in the device.</p> <p>HS Number of URB Entries must be divisible by 8 if the HS URB Entry Allocation Size is less than 9 512-bit URB entries. "2:0" = reserved "000b"</p>	Format:	U16	Value	Name	[0,1548]	
	Format:	U16						
Value	Name							
[0,1548]								
15:0	<p><b>HS Number of URB Entries Slice0</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>This field specifies the number of URB entries in Slice0 URB memory to be allocated to HS. SW shall ensure that the total HS Number of Entries does not exceed the relevant ValidValue range listed below.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 60%;">Value</th> <th style="width: 40%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,1548]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>HS Number of URB Entries must be divisible by 8 if the HS URB Entry Allocation Size is less than 9 512-bit URB entries. "2:0" = reserved "000b"</p>	Format:	U16	Value	Name	[0,1548]		
Format:	U16							
Value	Name							
[0,1548]								

## 3DSTATE\_URB\_ALLOC\_VS\_BODY

3DSTATE_URB_ALLOC_VS_BODY				
Source:	RenderCS			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:29	<b>Reserved</b>		
		Access: RO		
		Format: MBZ		
	28:21	<b>VS URB Starting Address SliceN</b>		
		Format: U8		
This field specifies the offset (from the start of the URB memory in additional slices) of the VS URB allocation, specified in multiples of 8 KB. For each additional, enabled slice, HW will increase the VS URB Starting Address by the value specified, resulting in a Starting Address within the total URB space.				
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,127]</td> <td></td> </tr> </tbody> </table>		Value	Name	[0,127]
Value	Name			
[0,127]				
<b>Programming Notes</b>				
This field is ignored by HW as there is only one slice in the device.				
20:18	<b>Reserved</b>			
	Access: RO			
	Format: MBZ			
17:10	<b>VS URB Starting Address Slice0</b>			
	Format: U8			
	This field specifies the offset (from the start of Slice0 URB memory) of the VS URB allocation, specified in multiples of 8 KB. This allocation must account for any Push Constant allocations, as those allocations begin at offset 0 in Slice0 URB memory.			
	<table border="1"> <thead> <tr> <th>Value</th> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>[0,127]</td> <td>The VS URB Starting Address Slice 0 must be greater than the render and push constant space allocated using 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_GS, 3DSTATE_PUSH_CONSTANT_ALLOC_PS and VSR_PUSH_CONSTANT_BASE.</td> </tr> </tbody> </table>	Value	Programming Notes	[0,127]
Value	Programming Notes			
[0,127]	The VS URB Starting Address Slice 0 must be greater than the render and push constant space allocated using 3DSTATE_PUSH_CONSTANT_ALLOC_VS, 3DSTATE_PUSH_CONSTANT_ALLOC_HS, 3DSTATE_PUSH_CONSTANT_ALLOC_DS, 3DSTATE_PUSH_CONSTANT_ALLOC_GS, 3DSTATE_PUSH_CONSTANT_ALLOC_PS and VSR_PUSH_CONSTANT_BASE.			
9:0	<b>VS URB Entry Allocation Size</b>			
	Format: U10-1			
Specifies the length, count of 512-bit units, of each URB entry owned by VS. This field is always used (even if VS Function Enable is DISABLED).				

<b>3DSTATE_URB_ALLOC_VS_BODY</b>																					
		<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>[0,511]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,511]																
Value	Name																				
[0,511]																					
		<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">As the VS URB entry serves as both the per-vertex input and output of the VS shader, the VS URB Allocation Size shall be sized to the maximum of the vertex input and output structures.</td> </tr> </tbody> </table>	Programming Notes		As the VS URB entry serves as both the per-vertex input and output of the VS shader, the VS URB Allocation Size shall be sized to the maximum of the vertex input and output structures.																
Programming Notes																					
As the VS URB entry serves as both the per-vertex input and output of the VS shader, the VS URB Allocation Size shall be sized to the maximum of the vertex input and output structures.																					
1	31:16	<p><b>VS Number of URB Entries SliceN</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U16</td> </tr> </table> <p>This field specifies the number of URB entries in slices beyond Slice0 to be allocated to VS. SW shall ensure that the total Number of Entries does not exceed the relevant ValidValue range listed below.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%; text-align: center;">Value</th> <th style="width: 30%; text-align: center;">Name</th> <th style="width: 40%;"></th> </tr> </thead> <tbody> <tr> <td>[64,3576]</td> <td></td> <td>RenderCS</td> </tr> <tr> <td>[64,1280]</td> <td></td> <td>PositionCS</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 field is ignored by HW as there is only one slice in the device.</td> </tr> <tr> <td colspan="2">VS URB entries shall be allocated even if VS Function Enable is DISABLED.</td> </tr> <tr> <td colspan="2">VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"</td> </tr> </tbody> </table>	Format:	U16	Value	Name		[64,3576]		RenderCS	[64,1280]		PositionCS	Programming Notes		This field is ignored by HW as there is only one slice in the device.		VS URB entries shall be allocated even if VS Function Enable is DISABLED.		VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"	
Format:	U16																				
Value	Name																				
[64,3576]		RenderCS																			
[64,1280]		PositionCS																			
Programming Notes																					
This field is ignored by HW as there is only one slice in the device.																					
VS URB entries shall be allocated even if VS Function Enable is DISABLED.																					
VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"																					
	15:0	<p><b>VS Number of URB Entries Slice0</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U16</td> </tr> </table> <p>This field specifies the number of URB entries in Slice0 URB memory to be allocated to VS. SW shall ensure that the total Number of Entries does not exceed the relevant ValidValue range listed below.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 30%; text-align: center;">Value</th> <th style="width: 30%; text-align: center;">Name</th> <th style="width: 40%;"></th> </tr> </thead> <tbody> <tr> <td>[64,3576]</td> <td></td> <td>RenderCS</td> </tr> <tr> <td>[64,1280]</td> <td></td> <td>PositionCS</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">VS URB entries shall be allocated even if VS Function Enable is DISABLED.</td> </tr> <tr> <td colspan="2">VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"</td> </tr> </tbody> </table>	Format:	U16	Value	Name		[64,3576]		RenderCS	[64,1280]		PositionCS	Programming Notes		VS URB entries shall be allocated even if VS Function Enable is DISABLED.		VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"			
Format:	U16																				
Value	Name																				
[64,3576]		RenderCS																			
[64,1280]		PositionCS																			
Programming Notes																					
VS URB entries shall be allocated even if VS Function Enable is DISABLED.																					
VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"																					

## 3DSTATE\_URB\_DS\_BODY

3DSTATE_URB_DS_BODY						
Source:	RenderCS					
Size (in bits):	32					
Default Value:	0x00000000					
DWord	Bit	Description				
0	31:25	<b>DS URB Starting Address</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U7</td></tr></table> Offset from the start of the URB memory where DS starts its allocation, specified in multiples of 8 KB.		U7		
			U7			
		<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>	Value	Name	[0,127]	
Value	Name					
[0,127]						
<p style="text-align: center;"><b>Programming Notes</b></p> If CTXT_SR_CTL::POSH_Enable is set and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 8. If CTXT_SR_CTL::POSH_Enable is clear and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 4. If Push Constants are not required and Device[SliceCount] == 1, the lower limit is 0.						
24:16		<b>DS URB Entry Allocation Size</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U9-1</td></tr></table> Specifies the length, count of 512-bit units, of each URB entry owned by DS. This field is always used (even if DS Function Enable is DISABLED).		U9-1		
			U9-1			
<b>DS Number of URB Entries</b> Specifies the number of URB entries that are used by DS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below. This field is always used (even if DS Function Enable is DISABLED). If Domain Shader Thread Dispatch is Enabled then the minimum number of handles that must be allocated is 34 URB entries.						
15:0		<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>[0,3576]</td> <td> </td> </tr> </tbody> </table>	Value	Name	[0,3576]	
		Value	Name			
		[0,3576]				
<p style="text-align: center;"><b>Programming Notes</b></p> DS Number of URB Entries must be divisible by 8 if the DS URB Entry Allocation Size is programmed to a value less than 9, which is 10 512-bit URB entries. "2:0" = reserved "000"						

## 3DSTATE\_URB\_GS\_BODY

3DSTATE_URB_GS_BODY						
Source:	RenderCS					
Size (in bits):	32					
Default Value:	0x00000000					
DWord	Bit	Description				
0	31:25	<b>GS URB Starting Address</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td></td><td>U7</td></tr></table> Offset from the start of the URB memory where GS starts its allocation, specified in multiples of 8 KB.		U7		
			U7			
		<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>	Value	Name	[0,127]	
Value	Name					
[0,127]						
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>           If CTXT_SR_CTL::POSH_Enable is set and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 8.            If CTXT_SR_CTL::POSH_Enable is clear and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 4.            If Push Constants are not required and Device[SliceCount] == 1, the lower limit is 0.         </td> </tr> </tbody> </table>	Programming Notes	If CTXT_SR_CTL::POSH_Enable is set and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 8. If CTXT_SR_CTL::POSH_Enable is clear and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 4. If Push Constants are not required and Device[SliceCount] == 1, the lower limit is 0.				
Programming Notes						
If CTXT_SR_CTL::POSH_Enable is set and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 8. If CTXT_SR_CTL::POSH_Enable is clear and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 4. If Push Constants are not required and Device[SliceCount] == 1, the lower limit is 0.						
24:16		<b>GS URB Entry Allocation Size</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td></td><td>U9-1</td></tr></table> Specifies the length, count of 512-bit units, of each URB entry owned by GS. This field is always used (even if GS Function Enable is DISABLED).		U9-1		
			U9-1			
<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>[0,1548]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,1548]			
Value	Name					
[0,1548]						
15:0		<b>GS Number of URB Entries</b> Specifies the number of URB entries that are used by GS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below. This field is always used (even if GS Function Enable is DISABLED).				
		<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>[0,1548]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,1548]	
		Value	Name			
		[0,1548]				
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>           Only if GS is disabled can this field be programmed to 0. If GS is enabled this field shall be programmed to a value greater than 0. For GS Dispatch Mode "Single", this field shall be programmed to a value greater than or equal to 1. For other GS Dispatch Modes, refer to the definition of Dispatch Mode (3DSTATE_GS) for minimum values of this field.             GS Number of URB Entries must be divisible by 8 if the GS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000"             When 3DSTATE_GS:Enable is true, the minimum number of GS Number of URB Entries must be set to 2.         </td> </tr> </tbody> </table>	Programming Notes	Only if GS is disabled can this field be programmed to 0. If GS is enabled this field shall be programmed to a value greater than 0. For GS Dispatch Mode "Single", this field shall be programmed to a value greater than or equal to 1. For other GS Dispatch Modes, refer to the definition of Dispatch Mode (3DSTATE_GS) for minimum values of this field.  GS Number of URB Entries must be divisible by 8 if the GS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000"  When 3DSTATE_GS:Enable is true, the minimum number of GS Number of URB Entries must be set to 2.				
Programming Notes						
Only if GS is disabled can this field be programmed to 0. If GS is enabled this field shall be programmed to a value greater than 0. For GS Dispatch Mode "Single", this field shall be programmed to a value greater than or equal to 1. For other GS Dispatch Modes, refer to the definition of Dispatch Mode (3DSTATE_GS) for minimum values of this field.  GS Number of URB Entries must be divisible by 8 if the GS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000"  When 3DSTATE_GS:Enable is true, the minimum number of GS Number of URB Entries must be set to 2.						
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>           Only if GS is disabled can this field be programmed to 0. If GS is enabled this field shall be programmed to a value greater than 0. For GS Dispatch Mode "Single", this field shall be programmed to a value greater than or equal to 1. For other GS Dispatch Modes, refer to the definition of Dispatch Mode (3DSTATE_GS) for minimum values of this field.         </td> </tr> </tbody> </table>	Programming Notes	Only if GS is disabled can this field be programmed to 0. If GS is enabled this field shall be programmed to a value greater than 0. For GS Dispatch Mode "Single", this field shall be programmed to a value greater than or equal to 1. For other GS Dispatch Modes, refer to the definition of Dispatch Mode (3DSTATE_GS) for minimum values of this field.				
Programming Notes						
Only if GS is disabled can this field be programmed to 0. If GS is enabled this field shall be programmed to a value greater than 0. For GS Dispatch Mode "Single", this field shall be programmed to a value greater than or equal to 1. For other GS Dispatch Modes, refer to the definition of Dispatch Mode (3DSTATE_GS) for minimum values of this field.						

## 3DSTATE\_URB\_HS\_BODY

3DSTATE_URB_HS_BODY						
Source:	RenderCS					
Size (in bits):	32					
Default Value:	0x00000000					
DWord	Bit	Description				
0	31:25	<b>HS URB Starting Address</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U7</td></tr></table> Offset from the start of the URB memory where HS starts its allocation, specified in multiples of 8 KB.		U7		
			U7			
		<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,127]</td> <td> </td> </tr> </tbody> </table>	Value	Name	[0,127]	
Value	Name					
[0,127]						
<p style="text-align: center;"><b>Programming Notes</b></p> If CTXT_SR_CTL::POSH_Enable is set and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 8. If CTXT_SR_CTL::POSH_Enable is clear and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 4. If Push Constants are not required and Device[SliceCount] == 1, the lower limit is 0.						
15:0	24:16	<b>HS URB Entry Allocation Size</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U9-1</td></tr></table> Specifies the length, count of 512-bit units, of each URB entry owned by HS. This field is always used (even if HS Function Enable is DISABLED).		U9-1		
			U9-1			
<b>HS Number of URB Entries</b> Specifies the number of URB entries that are used by HS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below. This field is always used (even if HS Function Enable is DISABLED). Programming Restriction:HS Number of URB Entries must be divisible by 8 if the HS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000"						
		<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,1548]</td> <td> </td> </tr> </tbody> </table>	Value	Name	[0,1548]	
Value	Name					
[0,1548]						

## 3DSTATE\_URB\_VS\_BODY

3DSTATE_URB_VS_BODY									
Source:	RenderCS								
Size (in bits):	32								
Default Value:	0x00000000								
DWord	Bit	Description							
0	31:25	<b>VS URB Starting Address</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U7</td></tr></table> Offset from the start of the URB memory where VS starts its allocation, specified in multiples of 8 KB.		U7					
			U7						
		<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>	Value	Name	[0,127]				
Value	Name								
[0,127]									
<p style="text-align: center;"><b>Programming Notes</b></p> If CTXT_SR_CTL::POSH_Enable is set and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 8. If CTXT_SR_CTL::POSH_Enable is clear and Push Constants are required or Device[SliceCount] GT 1, the lower limit is 4. If Push Constants are not required and Device[SliceCount] == 1, the lower limit is 0.									
24:16		<b>VS URB Entry Allocation Size</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U9-1</td></tr></table> Specifies the length, count of 512-bit units, of each URB entry owned by VS. This field is always used (even if VS Function Enable is DISABLED).		U9-1					
			U9-1						
		<p style="text-align: center;"><b>Programming Notes</b></p> Programming Restriction: As the VS URB entry serves as both the per-vertex input and output of the VS shader, the VS URB Allocation Size must be sized to the maximum of the vertex input and output structures.							
15:0		<b>VS Number of URB Entries</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U16</td></tr></table> Specifies the number of URB entries that are used by VS, based on only 1 slice enabled. When multiple slices are enabled, HW will multiply the value programmed by the number of slices in order to determine the total number of entries. SW shall ensure that the total number of entries does not exceed the relevant ValidValue range listed below. This field is always used (even if VS Function Enable is DISABLED).		U16					
			U16						
		<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 40%;"> </th> </tr> </thead> <tbody> <tr> <td>[64,3576]</td> <td> </td> <td>RenderCS</td> </tr> <tr> <td>[64,1280]</td> <td> </td> <td>PositionCS</td> </tr> </tbody> </table>	Value	Name		[64,3576]		RenderCS	[64,1280]
Value	Name								
[64,3576]		RenderCS							
[64,1280]		PositionCS							



<b>3DSTATE_URB_VS_BODY</b>		
		<b>Programming Notes</b>
		Programming Restriction: VS Number of URB Entries must be divisible by 8 if the VS URB Entry Allocation Size is less than 9 512-bit URB entries."2:0" = reserved "000b"

## 3DSTATE\_VF\_BODY

3DSTATE_VF_BODY		
Source:	RenderCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:0	<p><b>Cut Index</b></p> <p>This field specifies the index value that is considered the "cut index" which vertex indices are compared to if a Cut Index Enable is set. The Cut Index is compared to the fetched (and possibly-sign-extended) vertex index, and if these values are equal, the current primitive topology is terminated. Note that, for index buffers less than 32bpp, it is possible to set the Cut Index to a (large) value that will never match a sign-extended vertex index.</p>

## 3DSTATE\_VF\_COMPONENT\_PACKING\_BODY

3DSTATE_VF_COMPONENT_PACKING_BODY		
Source:	RenderCS	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..3	127:0	<b>Vertex Elements Enables</b>
		Format: <b>COMPONENT_ENABLES[32]</b>

## 3DSTATE\_VF\_INSTANCING\_BODY

3DSTATE_VF_INSTANCING_BODY					
Source:	RenderCS				
Size (in bits):	64				
Default Value:	0x00000000, 0x00000000				
DWord	Bit	Description			
0	31:10	<b>Reserved</b>			
		Access:	RO		
		Format:	MBZ		
9		<b>Instance Stride Enable</b>			
		Format:	Enable		
		<b>Value</b>	<b>Name</b>	<b>Description</b>	
		0h	Disabled	The Instance Stride value is neither defined nor used. For this vertex element, VF will access the Vertex Buffer as a simple 1D array using the Vertex Buffer Pitch as a stride.	
		1h	Enable	For this vertex element, VF will access the Vertex Buffer as a 2D array. The Instance Advancement State field defines the stride between instances. The Vertex Buffer Pitch defines the stride between vertices within a given instance.	
		<b>Programming Notes</b>			
		This field must be DISABLED when Instancing Enable is ENABLED.			
		8		<b>Instancing Enable</b>	
				Format:	Enable
				<b>Value</b>	<b>Name</b>
0h	Disabled			This vertex element is not instanced and therefore vertices within instances can each receive different data for this vertex element. Within each instance, the source vertex data for this vertex element is determined according the Vertex Access Type of the 3DPRIMITIVE command. There is no Instance Data Step Rate state defined for this vertex element.	
1h	Enabled			This vertex element is instanced and therefore vertices within instances will receive the same data for this vertex element. The source pointer for this particular vertex element will be (a) initialized at the start of 3DPRIMITIVE processing, (b) held constant for all vertices within an instance, and (c) advanced between instances as a function of Instance Data Step Rate.	The Instance Advancement State field provides the Instance Data Step Rate.

<b>3DSTATE_VF_INSTANCING_BODY</b>											
	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> <tr> <td colspan="2">This field must be DISABLED when Instance Stride Enable is ENABLED.</td> </tr> </table>	<b>Programming Notes</b>		This field must be DISABLED when Instance Stride Enable is ENABLED.							
<b>Programming Notes</b>											
This field must be DISABLED when Instance Stride Enable is ENABLED.											
	<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">7:6</td> <td><b>Reserved</b></td> </tr> <tr> <td style="width: 5%;"></td> <td> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> </td> </tr> </table>	7:6	<b>Reserved</b>		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
7:6	<b>Reserved</b>										
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
Access:	RO										
Format:	MBZ										
	<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">5:0</td> <td><b>Vertex Element Index</b></td> </tr> <tr> <td style="width: 5%;"></td> <td> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U6</td> </tr> </table> <p>This field identifies which vertex element state is to be modified by this command.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,33]</td> <td></td> </tr> </tbody> </table> </td> </tr> </table>	5:0	<b>Vertex Element Index</b>		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U6</td> </tr> </table> <p>This field identifies which vertex element state is to be modified by this command.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,33]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,33]	
5:0	<b>Vertex Element Index</b>										
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U6</td> </tr> </table> <p>This field identifies which vertex element state is to be modified by this command.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,33]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,33]					
Format:	U6										
Value	Name										
[0,33]											
1	<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">31:0</td> <td> <p><b>Instance Advancement State</b></p> <p>If Instancing Enable is ENABLED, this field determines the rate at which data for this particular vertex element is changed between instances. Only after the number of instances specified by this field is generated is new (sequential) vertex element data provided. This process continues for each group of instances defined in the 3DPRIMITIVE command. For example, a value of 1 in this field causes new data to be supplied for this vertex element with each sequential (instance) group of vertices. A value of 2 causes every other instance group of vertices to be provided with new vertex element data. The special value of 0 causes all vertices of all instances generated by the 3DPRIMITIVE command to be provided with the same data for this vertex element. (The same effect can be achieved by setting this field to its maximum value.)</p> <p>If Instance Stride Enable is ENABLED, this field determines the stride in BYTES from one instance to the next. An InstanceStride of 0 means there is only one set of instance data (a degenerate 2D array with an instance dimension of 1). For this vertex element, VF will access the vertex buffer data sequentially for each vertex within first instance, and then return to the start of the vertex buffer for the vertices in the next instance, and so on.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> <tr> <td colspan="2">The interpretation/use of this field depends on whether the Instancing Enable or Instance Stride Enable state bit is ENABLED. Software shall not concurrently ENABLE both of those state bits. If both state bits are DISABLED, this field is ignored.</td> </tr> </table> </td> </tr> </table>	31:0	<p><b>Instance Advancement State</b></p> <p>If Instancing Enable is ENABLED, this field determines the rate at which data for this particular vertex element is changed between instances. Only after the number of instances specified by this field is generated is new (sequential) vertex element data provided. This process continues for each group of instances defined in the 3DPRIMITIVE command. For example, a value of 1 in this field causes new data to be supplied for this vertex element with each sequential (instance) group of vertices. A value of 2 causes every other instance group of vertices to be provided with new vertex element data. The special value of 0 causes all vertices of all instances generated by the 3DPRIMITIVE command to be provided with the same data for this vertex element. (The same effect can be achieved by setting this field to its maximum value.)</p> <p>If Instance Stride Enable is ENABLED, this field determines the stride in BYTES from one instance to the next. An InstanceStride of 0 means there is only one set of instance data (a degenerate 2D array with an instance dimension of 1). For this vertex element, VF will access the vertex buffer data sequentially for each vertex within first instance, and then return to the start of the vertex buffer for the vertices in the next instance, and so on.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> <tr> <td colspan="2">The interpretation/use of this field depends on whether the Instancing Enable or Instance Stride Enable state bit is ENABLED. Software shall not concurrently ENABLE both of those state bits. If both state bits are DISABLED, this field is ignored.</td> </tr> </table>	<b>Programming Notes</b>		The interpretation/use of this field depends on whether the Instancing Enable or Instance Stride Enable state bit is ENABLED. Software shall not concurrently ENABLE both of those state bits. If both state bits are DISABLED, this field is ignored.					
31:0	<p><b>Instance Advancement State</b></p> <p>If Instancing Enable is ENABLED, this field determines the rate at which data for this particular vertex element is changed between instances. Only after the number of instances specified by this field is generated is new (sequential) vertex element data provided. This process continues for each group of instances defined in the 3DPRIMITIVE command. For example, a value of 1 in this field causes new data to be supplied for this vertex element with each sequential (instance) group of vertices. A value of 2 causes every other instance group of vertices to be provided with new vertex element data. The special value of 0 causes all vertices of all instances generated by the 3DPRIMITIVE command to be provided with the same data for this vertex element. (The same effect can be achieved by setting this field to its maximum value.)</p> <p>If Instance Stride Enable is ENABLED, this field determines the stride in BYTES from one instance to the next. An InstanceStride of 0 means there is only one set of instance data (a degenerate 2D array with an instance dimension of 1). For this vertex element, VF will access the vertex buffer data sequentially for each vertex within first instance, and then return to the start of the vertex buffer for the vertices in the next instance, and so on.</p> <table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> <tr> <td colspan="2">The interpretation/use of this field depends on whether the Instancing Enable or Instance Stride Enable state bit is ENABLED. Software shall not concurrently ENABLE both of those state bits. If both state bits are DISABLED, this field is ignored.</td> </tr> </table>	<b>Programming Notes</b>		The interpretation/use of this field depends on whether the Instancing Enable or Instance Stride Enable state bit is ENABLED. Software shall not concurrently ENABLE both of those state bits. If both state bits are DISABLED, this field is ignored.							
<b>Programming Notes</b>											
The interpretation/use of this field depends on whether the Instancing Enable or Instance Stride Enable state bit is ENABLED. Software shall not concurrently ENABLE both of those state bits. If both state bits are DISABLED, this field is ignored.											

## 3DSTATE\_VF\_SGVS\_2\_BODY

3DSTATE_VF_SGVS_2_BODY				
Source:		RenderCS		
Size (in bits):		64		
Default Value:		0x00000000, 0x00000000		
DWord	Bit	Description		
0	31	<b>XP1 Enable</b>		
		Format: Boolean		
		Value	Name	Description
		0h	Disabled	XP1 is not inserted
	1h	Enabled	XP1 (as defined by XP1 Source Select) is inserted.	
	30:29	<b>XP1 Component Number</b>		
		If XP1 Enable is ENABLED, this field specifies the 32-bit component location (within the 4-component VUE) where it is inserted.		
		If XP1 Enable is DISABLED, this field is ignored.		
		Value	Name	Description
		0	COMP_0	If enabled, XP1 is inserted in component 0 (.x)
1	COMP_1	If enabled, XP1 is inserted in component 1 (.y)		
2	COMP_2	If enabled, XP1 is inserted in component 2 (.z)		
3	COMP_3	If enabled, XP1 is inserted in component 3 (.w)		
28	<b>XP1 Source Select</b>			
	If XP1 Enable is ENABLED, this field selects between the available sources for the XP1 SGV to be inserted.			
	If XP1 Enable is DISABLED, this field is ignored.			
	Value	Name	Description	Programming Notes
1h	Start Instance Location	The XP1 value is sourced from the Start Instance Location Parameter.	Start Instance Location is the only valid value if 3DSTATE_VF::InstanceIDOffsetEnable is set.	
0h	XP1_PARAMETER	The XP1 value is sourced from the XP1 parameter as defined by 3DPRIMITIVE.		
27:22	<b>Reserved</b>			
	Access:		RO	
	Format:		MBZ	

## 3DSTATE\_VF\_SGVS\_2\_BODY

21:16	<b>XP1 Element Offset</b>		
	Format:	U6	
	If XP1 Enable is ENABLED, this field specifies the VUE element offset of the 128-bit element where it is to be inserted. The XP1 Component Number specifies where in the specified element it is to be inserted.		
	<b>Value</b>	<b>Name</b>	
	[0,33]		
15	<b>XP0 Enable</b>		
	Format:	Boolean	
	<b>Value</b>	<b>Name</b>	
	<b>Description</b>		
	0h	Disabled	XP0 is not inserted
	1h	Enabled	XP0 (as defined by XP0 Source Select) is inserted
14:13	<b>XP0 Component Number</b>		
	If XP0 Enable is ENABLED, this field specifies the 32-bit component location (within the 4-component VUE) where it is inserted. If XP0 Enable is DISABLED, this field is ignored.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0	COMP_0	If enabled, XP0 is inserted in component 0 (.x)
	1	COMP_1	If enabled, XP0 is inserted in component 1 (.y)
	2	COMP_2	If enabled, XP0 is inserted in component 2 (.z)
3	COMP_3	If enabled, XP0 is inserted in component 3 (.w)	
12	<b>XP0 Source Select</b>		
	If XP0 Enable is ENABLED, this field selects between the available sources for the XP0 SGV to be inserted.		
	If XP0 Enable is DISABLED, this field is ignored.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	1h	VERTEX_LOCATION	The XP0 value is sourced from one of the two Vertex Location parameters passed in 3DPRIMITIVE. If Vertex Access Mode is SEQUENTIAL, the Start Vertex Location value is used. If Vertex Access Mode is RANDOM, the Base Vertex Location value is used.
	0h	XP0_PARAMETER	The XP0 value is sourced from the XP0 parameter as defined by 3DPRIMITIVE.
11:6	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	

### 3DSTATE\_VF\_SGVS\_2\_BODY

	5:0	<b>XP0 Element Offset</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>If XP0 Enable is ENABLED, this field specifies the VUE element offset of the 128-bit element where it is to be inserted. The XP0 Component Number specifies where in the specified element it is to be inserted. If XP0 Enable is DISABLED, this field is ignored.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,33]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,33]										
Format:	U6																	
Value	Name																	
[0,33]																		
1	31:16	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="width: 40%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ											
Access:	RO																	
Format:	MBZ																	
	15	<b>XP2 Enable</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Boolean</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 10%;">Value</th> <th style="width: 10%;">Name</th> <th style="width: 80%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disabled</td> <td>XP2 is not inserted</td> </tr> <tr> <td>1h</td> <td>Enabled</td> <td>XP2 is inserted, sourced from the XP2 parameter as defined by 3DPRIMITIVE.</td> </tr> </tbody> </table>	Format:	Boolean	Value	Name	Description	0h	Disabled	XP2 is not inserted	1h	Enabled	XP2 is inserted, sourced from the XP2 parameter as defined by 3DPRIMITIVE.				
Format:	Boolean																	
Value	Name	Description																
0h	Disabled	XP2 is not inserted																
1h	Enabled	XP2 is inserted, sourced from the XP2 parameter as defined by 3DPRIMITIVE.																
	14:13	<b>XP2 Component Number</b>	<p>If XP2 Enable is ENABLED, this field specifies the 32-bit component location (within the 4-component VUE) where it is to be inserted. If XP2 Enable is DISABLED, this field is ignored.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <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>0</td> <td>COMP_0</td> <td>If enabled, XP2 is inserted in component 0 (.x)</td> </tr> <tr> <td>1</td> <td>COMP_1</td> <td>If enabled, XP2 is inserted in component 1 (.y)</td> </tr> <tr> <td>2</td> <td>COMP_2</td> <td>If enabled, XP2 is inserted in component 2 (.z)</td> </tr> <tr> <td>3</td> <td>COMP_3</td> <td>If enabled, XP2 is inserted in component 3 (.w)</td> </tr> </tbody> </table>	Value	Name	Description	0	COMP_0	If enabled, XP2 is inserted in component 0 (.x)	1	COMP_1	If enabled, XP2 is inserted in component 1 (.y)	2	COMP_2	If enabled, XP2 is inserted in component 2 (.z)	3	COMP_3	If enabled, XP2 is inserted in component 3 (.w)
Value	Name	Description																
0	COMP_0	If enabled, XP2 is inserted in component 0 (.x)																
1	COMP_1	If enabled, XP2 is inserted in component 1 (.y)																
2	COMP_2	If enabled, XP2 is inserted in component 2 (.z)																
3	COMP_3	If enabled, XP2 is inserted in component 3 (.w)																
	12:6	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="width: 40%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ											
Access:	RO																	
Format:	MBZ																	
	5:0	<b>XP2 Element Offset</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U6</td> </tr> </table> <p>If XP2 Enable is ENABLED, this field specifies the VUE element offset of the 128-bit element where it is to be inserted. The XP2 Component Number specifies where in the specified element it is to be inserted. If XP2 Enable is DISABLED, this field is ignored.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,33]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,33]										
Format:	U6																	
Value	Name																	
[0,33]																		



## 3DSTATE\_VF\_SGVS\_BODY

3DSTATE_VF_SGVS_BODY																	
Source:	RenderCS																
Size (in bits):	32																
Default Value:	0x00000000																
DWord	Bit	Description															
0	31	<b>InstancelD Enable</b>															
		Format: Enable															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disabled</td> <td>InstancelD is not inserted</td> </tr> <tr> <td>1h</td> <td>Enabled</td> <td>InstancelD is inserted</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disabled	InstancelD is not inserted	1h	Enabled	InstancelD is inserted						
		Value	Name	Description													
	0h	Disabled	InstancelD is not inserted														
	1h	Enabled	InstancelD is inserted														
	30:29	<b>InstancelD Component Number</b>															
	<p>If InstancelD Enable is ENABLED, this field specifies the 32-bit component location (within the 4-component VUE) where it is inserted.</p> <p>If InstancelD Enable is DISABLED, this field is ignored.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>COMP_0</td> <td>If enabled, InstancelD is inserted in component 0 (.x)</td> </tr> <tr> <td>1</td> <td>COMP_1</td> <td>If enabled, InstancelD is inserted in component 1 (.y)</td> </tr> <tr> <td>2</td> <td>COMP_2</td> <td>If enabled, InstancelD is inserted in component 2 (.z)</td> </tr> <tr> <td>3</td> <td>COMP_3</td> <td>If enabled, InstancelD is inserted in component 3 (.w)</td> </tr> </tbody> </table>		Value	Name	Description	0	COMP_0	If enabled, InstancelD is inserted in component 0 (.x)	1	COMP_1	If enabled, InstancelD is inserted in component 1 (.y)	2	COMP_2	If enabled, InstancelD is inserted in component 2 (.z)	3	COMP_3	If enabled, InstancelD is inserted in component 3 (.w)
	Value	Name	Description														
0	COMP_0	If enabled, InstancelD is inserted in component 0 (.x)															
1	COMP_1	If enabled, InstancelD is inserted in component 1 (.y)															
2	COMP_2	If enabled, InstancelD is inserted in component 2 (.z)															
3	COMP_3	If enabled, InstancelD is inserted in component 3 (.w)															
28:22	<b>Reserved</b>																
Access: RO																	
Format: MBZ																	
21:16	<b>InstancelD Element Offset</b>																
	Format: U6																
	<p>If InstancelD Enable is ENABLED, this field specifies the VUE element offset of the 128-bit element where it is to be inserted. The InstancelD Component Number specifies where in the specified element it is inserted.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,33]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,33]													
Value	Name																
[0,33]																	
15	<b>VertexID Enable</b>																
	Format: Enable																
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disabled</td> <td>VertexID is not inserted</td> </tr> <tr> <td>1h</td> <td>Enabled</td> <td>VertexID is inserted</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disabled	VertexID is not inserted	1h	Enabled	VertexID is inserted							
	Value	Name	Description														
0h	Disabled	VertexID is not inserted															
1h	Enabled	VertexID is inserted															

### 3DSTATE\_VF\_SGVS\_BODY

14:13		<b>VertexID Component Number</b>	<p>If VertexID Enable is ENABLED, this field specifies the 32-bit component location (within the 4-component VUE) where it is inserted. If VertexID Enable is DISABLED, this field is ignored.</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 style="text-align: center;">0</td> <td>COMP_0</td> <td>If enabled, VertexID is inserted in component 0 (.x)</td> </tr> <tr> <td style="text-align: center;">1</td> <td>COMP_1</td> <td>If enabled, VertexID is inserted in component 1 (.y)</td> </tr> <tr> <td style="text-align: center;">2</td> <td>COMP_2</td> <td>If enabled, VertexID is inserted in component 2 (.z)</td> </tr> <tr> <td style="text-align: center;">3</td> <td>COMP_3</td> <td>If enabled, VertexID is inserted in component 3 (.w)</td> </tr> </tbody> </table>	Value	Name	Description	0	COMP_0	If enabled, VertexID is inserted in component 0 (.x)	1	COMP_1	If enabled, VertexID is inserted in component 1 (.y)	2	COMP_2	If enabled, VertexID is inserted in component 2 (.z)	3	COMP_3	If enabled, VertexID is inserted in component 3 (.w)
Value	Name	Description																
0	COMP_0	If enabled, VertexID is inserted in component 0 (.x)																
1	COMP_1	If enabled, VertexID is inserted in component 1 (.y)																
2	COMP_2	If enabled, VertexID is inserted in component 2 (.z)																
3	COMP_3	If enabled, VertexID is inserted in component 3 (.w)																
12:6		<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ											
Access:	RO																	
Format:	MBZ																	
5:0		<b>VertexID Element Offset</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td>U6</td> </tr> </table> <p>If VertexID Enable is ENABLED, this field specifies the VUE element offset of the 128-bit element where it is to be inserted. The VertexID Component Number specifies where in the specified element it is inserted. This is also the vertex element index. If VertexID Enable is DISABLED, this field is ignored.</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>[0,33]</td> <td></td> </tr> </tbody> </table>	Format:	U6	Value	Name	[0,33]										
Format:	U6																	
Value	Name																	
[0,33]																		

## 3DSTATE\_VF\_TOPOLOGY\_BODY

3DSTATE_VF_TOPOLOGY_BODY		
Source:	RenderCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:6	<b>Reserved</b>
		Access: RO
		Format: MBZ
	5:0	<b>Primitive Topology Type</b> Format: <b>3D_Prim_Topo_Type</b> This field specifies the VF stage's Topology state.



## 3DSTATE\_VIEWPORT\_STATE\_POINTERS\_CC\_BODY

3DSTATE_VIEWPORT_STATE_POINTERS_CC_BODY		
Source:	RenderCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:5	<b>CC Viewport Pointer</b> Format: DynamicStateOffset[31:5]CC_VIEWPORT*16 Specifies the 32-byte aligned address offset of the CC_VIEWPORT state. This offset is relative to the Dynamic State Base Address.
	4:0	<b>Reserved</b> Access: RO Format: MBZ

## 3DSTATE\_VIEWPORT\_STATE\_POINTERS\_SF\_CLIP\_BODY

3DSTATE_VIEWPORT_STATE_POINTERS_SF_CLIP_BODY					
Source:	RenderCS				
Size (in bits):	32				
Default Value:	0x00000000				
DWord	Bit	Description			
0	31:6	<p><b>SF Clip Viewport Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>DynamicStateOffset[31:6]SF_CLIP_VIEWPORT*16</td> </tr> </table> <p>Specifies the 64-byte aligned address offset of the SF_CLIP_VIEWPORT state. This offset is relative to the Dynamic State Base Address.</p>	Format:	DynamicStateOffset[31:6]SF_CLIP_VIEWPORT*16	
	Format:	DynamicStateOffset[31:6]SF_CLIP_VIEWPORT*16			
5:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				

## 3DSTATE\_VS\_BODY

<b>3DSTATE_VS_BODY</b>													
Source:	RenderCS												
Size (in bits):	256												
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000												
DWord	Bit	Description											
0..1	63:6	<p><b>Kernel Start Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[63:6]</td> </tr> </table> <p>This field specifies the starting location of the kernel program run by threads spawned by the VS pipeline stage. It is specified as a 64-byte-granular offset from the Instruction Base Address. This field is ignored if VS Function Enable is DISABLED.</p>	Format:	InstructionBaseOffset[63:6]									
	Format:	InstructionBaseOffset[63:6]											
5:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
Access:	RO												
Format:	MBZ												
2	31	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ							
	Access:	RO											
Format:	MBZ												
	30	<p><b>Vector Mask Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Upon subsequent VS thread dispatches, this bit is loaded into the EUs <b>Vector Mask Enable</b> (VME, cr0.0[3]) thread state. Refer to EU documentation for the definition and use of VME state.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Dmask</td> <td>The EU will use the Dispatch Mask (supplied by the VS stage) for instruction execution.</td> </tr> <tr> <td>1h</td> <td>Vmask</td> <td>The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>Under normal conditions SW shall specify DMask, as the VS stage will provide a Dispatch Mask appropriate to SIMD4x2 or SIMD8 thread execution (as a function of SIMD8 Dispatch Enable). E.g., for SIMD4x2 thread execution, the VS stage will generate a Dispatch Mask that is equal to what the EU would use as the Vector Mask. For SIMD8 execution there is no known usage model for use of Vector Mask (as there is for PS shaders).</p>	Format:	Enable	Value	Name	Description	0h	Dmask	The EU will use the Dispatch Mask (supplied by the VS stage) for instruction execution.	1h	Vmask	The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.
Format:	Enable												
Value	Name	Description											
0h	Dmask	The EU will use the Dispatch Mask (supplied by the VS stage) for instruction execution.											
1h	Vmask	The EU will use the Vector Mask (derived from the Dispatch Mask) for instruction execution.											

## 3DSTATE\_VS\_BODY

	29:27	<p><b>Sampler Count</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U3</td> </tr> </table> <p>This field specifies (in multiples of 4) the number of sets of sampler state that will be prefetched for use by the VS kernel. While the prefetching of sampler state is optional and does not impact functionality, it may improve performance.</p> <p>This field is ignored if the Function Enable state is set to DISABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No Samplers</td> <td>no samplers used</td> </tr> <tr> <td>1h</td> <td>1-4 Samplers</td> <td>between 1 and 4 samplers used</td> </tr> <tr> <td>2h</td> <td>5-8 Samplers</td> <td>between 5 and 8 samplers used</td> </tr> <tr> <td>3h</td> <td>9-12 Samplers</td> <td>between 9 and 12 samplers used</td> </tr> <tr> <td>4h</td> <td>13-16 Samplers</td> <td>between 13 and 16 samplers used</td> </tr> </tbody> </table>	Format:	U3	Value	Name	Description	0h	No Samplers	no samplers used	1h	1-4 Samplers	between 1 and 4 samplers used	2h	5-8 Samplers	between 5 and 8 samplers used	3h	9-12 Samplers	between 9 and 12 samplers used	4h	13-16 Samplers	between 13 and 16 samplers used
Format:	U3																					
Value	Name	Description																				
0h	No Samplers	no samplers used																				
1h	1-4 Samplers	between 1 and 4 samplers used																				
2h	5-8 Samplers	between 5 and 8 samplers used																				
3h	9-12 Samplers	between 9 and 12 samplers used																				
4h	13-16 Samplers	between 13 and 16 samplers used																				
	26	<p><b>Reserved</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																
Access:	RO																					
Format:	MBZ																					
	25:18	<p><b>Binding Table Entry Count</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</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. Note: For kernels using a large number of binding table entries, it may be wise to set this field to zero to avoid prefetching too many entries and thrashing the state cache. This field is ignored if VS Function Enable is DISABLED.</p> <p><b>When HW Generated Binding Table bit is enabled:</b> This field indicates which cache lines (512bit units - 32 Binding Table Entry section) should be fetched. Each bit in this field corresponds to a cache line. Only the 1st 4 non-zero Binding Table entries of each 32 Binding Table entry section prefetched will have its surface state prefetched.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,255]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>When HW binding table bit is set, it is assumed that the Binding Table Entry Count field will be generated at JIT time.</p>	Format:	U8	Value	Name	[0,255]															
Format:	U8																					
Value	Name																					
[0,255]																						
	17	<p><b>Thread Dispatch Priority</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <p>Specifies the priority of the thread for dispatch: This field is ignored if VS Function Enable is DISABLED.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Normal</td> <td>Normal Priority</td> </tr> <tr> <td>1h</td> <td>High</td> <td>High Priority</td> </tr> </tbody> </table>	Format:	U1	Value	Name	Description	0h	Normal	Normal Priority	1h	High	High Priority									
Format:	U1																					
Value	Name	Description																				
0h	Normal	Normal Priority																				
1h	High	High Priority																				

## 3DSTATE\_VS\_BODY

16	<b>Floating Point Mode</b>		
	Format:	U1	
	Specifies the initial floating point mode used by the dispatched thread. This field is ignored if VS Function Enable is DISABLED.		
	<b>Value</b>	<b>Name</b>	
	0h	IEEE-754	
	1h	Alternate	
	<b>Description</b>		
	Use IEEE-754 Rules		
	Use Alternate Rules		
	15	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	14	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	13	<b>Illegal Opcode Exception Enable</b>	
Format:		Enable	
This bit gets loaded into EU CR0.1[12] (note the bit # difference). See Exceptions and ISA Execution Environment. This field is ignored if VS Function Enable is DISABLED.			
12	<b>Accesses UAV</b>		
	Format:	Enable	
	This field must be set when VS has a UAV access.		
	<b>Programming Notes</b>		
	This field must not be set when VS Function Enable is disabled.		
	This bit shall not be set when the command is executed in the PCS pipeline.		
11	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
10:8	<b>Reserved</b>		
	Format:	MBZ	
7	<b>Software Exception Enable</b>		
	Format:	Enable	
	This bit gets loaded into EU CR0.1[13] (note the bit # difference). See Exceptions and ISA Execution Environment. This field is ignored if VS Function Enable is DISABLED.		
6:0	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	



<b>3DSTATE_VS_BODY</b>							
3..4	63:32	<b>Reserved</b>					
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO					
	Format:	MBZ					
	31:10	<b>Scratch Space Base Pointer</b>					
Format:		GeneralStateOffset[31:10]					
<p>Specifies the starting location of the scratch space area allocated to this FF unit as a 1K-byte aligned offset from the General State Base Address. If required, each thread spawned by this FF unit will be allocated some portion of this space, as specified by Per-Thread Scratch Space. The computed offset of the thread-specific portion will be passed in the thread payload as Scratch Space Offset. The thread is expected to utilize "stateless" DataPort read/write requests to access scratch space, where the DataPort will cause the General State Base Address to be added to the offset passed in the request header. This field is ignored if VS Function Enable is DISABLED. In 64b OS all pointers need to be seen by SW as 48b. HW does not support a Scratch Space Base Pointer larger than 32b, therefore SW must ensure Bits&lt;63:32&gt; are set to 0's.</p>							
<b>Programming Notes</b>							
The scratch spaces allocated to the POCS VSR stage and RCS VS stage shall not overlap with each other or the scratch space allocations of any other enabled stage in the RCS pipeline.							
9:4	<b>Reserved</b>						
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
Access:	RO						
Format:	MBZ						
3:0	<b>Per-Thread Scratch Space</b>						
	Format:	U4					
	<p>Specifies the amount of scratch space to be allocated to each thread spawned by this FF unit. The driver must allocate enough contiguous scratch space, starting at the Scratch Space Base Pointer, to ensure that the Maximum Number of Threads can each get Per-Thread Scratch Space size without exceeding the driver-allocated scratch space. This field is ignored if VS Function Enable is DISABLED.</p>						
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 20%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0,11]</td> <td></td> <td>Indicating [1K Bytes, 2M Bytes]</td> </tr> </tbody> </table>	Value	Name	Description	[0,11]		Indicating [1K Bytes, 2M Bytes]
	Value	Name	Description				
[0,11]		Indicating [1K Bytes, 2M Bytes]					
<b>Programming Notes</b>							
This amount is available to the kernel for information only. It will be passed verbatim (if not altered by the kernel) to the Data Port in any scratch space access messages, but the Data Port will ignore it.							
5	31:25	<b>Reserved</b>					
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
		Access:	RO				
Format:	MBZ						

## 3DSTATE\_VS\_BODY

24:20	<b>Dispatch GRF Start Register For URB Data</b>										
	Format:	U5									
<p>Specifies the starting GRF number for the URB portion (URB constants and vertices) of the thread payload. This field is ignored if VS Function Enable is DISABLED.</p>											
<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>[0,31]</td> <td></td> <td>indicating GRF [R0, R31]</td> </tr> </tbody> </table>			Value	Name	Description	[0,31]		indicating GRF [R0, R31]			
Value	Name	Description									
[0,31]		indicating GRF [R0, R31]									
19:17	<b>Reserved</b>										
	Access:	RO									
Format: MBZ											
16:11	<b>Vertex URB Entry Read Length</b>										
	Format:	U6									
<p>Specifies the number of pairs of 128-bit vertex elements to be passed into the payload for each vertex. This field is ignored if VS Function Enable is DISABLED. For SIMD4x2 dispatch, each vertex element requires one GRF of payload data, therefore the number of GRFs with vertex data will be double the value programmed in this field. For SIMD8 dispatch, each vertex element requires 4 GRFs of payload data, therefore the number of GRFs with vertex data will be 8 times the value programmed in this field. The EU limit of 128 GRFs imposes a maximum limit of 30 elements per vertex pushed into the payload, though the practical limit may be lower. If input vertices exceed the practical limit, software must decide between resorting to pulling elements during thread execution or dropping back to SIMD4x2 dispatch. Note that the VUE is used for both input and output, so when using the pull-model software must ensure inputs are not overwritten before last use.</p>											
<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>[1,63]</td> <td></td> <td>if SIMD8 dispatch disabled</td> </tr> <tr> <td>[0,15]</td> <td></td> <td>if SIMD8 dispatch enabled</td> </tr> </tbody> </table>			Value	Name	Description	[1,63]		if SIMD8 dispatch disabled	[0,15]		if SIMD8 dispatch enabled
Value	Name	Description									
[1,63]		if SIMD8 dispatch disabled									
[0,15]		if SIMD8 dispatch enabled									
10	<b>Reserved</b>										
	Access:	RO									
Format: MBZ											
9:4	<b>Vertex URB Entry Read Offset</b>										
	Format:	U6									
<p>Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB before being included in the thread payload. This offset applies to all Vertex URB entries passed to the thread. This field is ignored if VS Function Enable is DISABLED.</p>											
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,63]</td> <td></td> </tr> </tbody> </table>			Value	Name	[0,63]						
Value	Name										
[0,63]											
3:0	<b>Reserved</b>										
	Access:	RO									
Format: MBZ											

## 3DSTATE\_VS\_BODY

6	31:22	<b>Maximum Number of Threads</b>								
		Format:	U10-1							
		<p>Specifies the maximum number of simultaneous threads allowed to be active. Used to avoid using up the scratch space. Programming the value of the max threads over the number of threads based off number of threads supported in the execution units may improve performance since the architecture allows threads to be buffered between the check for max threads and the actual dispatch into the EU. Programming the max values to a number less than the number of threads supported in the execution units may reduce performance. This field is ignored if VS Function Enable is DISABLED.</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;">[0,191]</td> <td></td> <td>indicating thread count of [1,192]</td> </tr> <tr> <td style="text-align: center;">[0,545]</td> <td></td> <td>indicating thread count of [1,546]</td> </tr> </tbody> </table>	Value	Name	Description	[0,191]		indicating thread count of [1,192]	[0,545]	
Value	Name	Description								
[0,191]		indicating thread count of [1,192]								
[0,545]		indicating thread count of [1,546]								
	21:11	<b>Reserved</b>								
		Access:	RO							
		Format:	MBZ							
	10	<b>Statistics Enable</b>								
		Format:	Enable							
		<p>If ENABLED, the VS stage will perform statistics gathering. See the Statistics Gathering subsection.</p> <p>If DISABLED, statistics information associated with the VS stage will be left unchanged.</p>								
		<b>Programming Notes</b>								
		<p>When a 3DPRIMITIVE command with POSH Enable set is executed from the RCS command stream, VS statistics gathering is inhibited for that command.</p>								
	9	<b>SIMD8 Single Instance Dispatch Enable</b>								
		Format:	Enable							
		<p>This field is used to specify whether vertices from different instances can be combined in a single SIMD8 dispatch. This bit is <u>ignored</u> if SIMD4x2 dispatches are enabled (i.e., SIMD8 Dispatch Enable is DISABLED).</p> <p>If ENABLED, SIMD8 VS thread dispatches <u>will not</u> combine vertices from different instances. This allows the VS kernel to handle instance-specific operations (e.g., read constants indexed by the InstanceID) in a global fashion, as these operations pertain to all vertices of the dispatch.</p> <p>If DISABLED, SIMD8 VS thread dispatches can combine vertices from different instances. The VS kernel must determine if instance-specific operations can be handled globally (vs. per-vertex). E.g., it can examine the Single Instance payload bit.</p>								
		<b>Programming Notes</b>								
		<p>SIMD8 Single Instance Dispatch Enable is not supported for HPCXTs.</p>								
	8:3	<b>Reserved</b>								
		Access:	RO							
		Format:	MBZ							

## 3DSTATE\_VS\_BODY

<b>3DSTATE_VS_BODY</b>										
2	<table border="1" style="width: 100%;"> <tr> <td colspan="2"><b>SIMD8 Dispatch Enable</b></td> </tr> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field determines how VS threads are dispatched and how the thread payloads are generated. The setting of this field must agree with how the VS kernel was compiled.</p> <p>If ENABLED, SIMD8 VS thread dispatches are performed. The <b>Single Vertex Dispatch</b> field is ignored.</p> <p>If DISABLED, SIMD4x2 thread dispatches are performed. The <b>Single Vertex Dispatch</b> field can be used to force single-vertex dispatches.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="2">The only supported mode is SIMD8 Dispatch Enable set to Enable (1).</td> </tr> </table>	<b>SIMD8 Dispatch Enable</b>		Format:	Enable	<b>Programming Notes</b>		The only supported mode is SIMD8 Dispatch Enable set to Enable (1).		
	<b>SIMD8 Dispatch Enable</b>									
	Format:	Enable								
<b>Programming Notes</b>										
The only supported mode is SIMD8 Dispatch Enable set to Enable (1).										
<table border="1" style="width: 100%;"> <tr> <td colspan="2"><b>Vertex Cache Disable</b></td> </tr> <tr> <td style="width: 60%;">Format:</td> <td>Disable</td> </tr> </table> <p>This bit controls the operation of the Vertex Cache. This field is always used.</p> <p>If the Vertex Cache is DISABLED and the VS Function is ENABLED, the Vertex Cache is not used and all incoming vertices will be passed to VS threads.</p> <p>If the Vertex Cache is ENABLED and the VS Function is ENABLED, only incoming vertices that do not hit in the Vertex Cache will be passed to VS threads.</p> <p>If the Vertex Cache is ENABLED and the VS Function is DISABLED, input vertices that miss in the Vertex Cache will be assembled and written to the URB (by the VF stage), and subsequently passed through the VS stage unmodified (i.e, no VS threads are spawned).</p> <p>The Vertex Cache is invalidated whenever the Vertex Cache becomes DISABLED, whenever the VS Function Enable toggles, between 3DPRIMITIVE commands and between instances within a 3DPRIMITIVE command.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="2">See the Vertex Caching subsection for details on implicit Vertex Cache disabling.</td> </tr> </table>	<b>Vertex Cache Disable</b>		Format:	Disable	<b>Programming Notes</b>		See the Vertex Caching subsection for details on implicit Vertex Cache disabling.			
<b>Vertex Cache Disable</b>										
Format:	Disable									
<b>Programming Notes</b>										
See the Vertex Caching subsection for details on implicit Vertex Cache disabling.										
0	<table border="1" style="width: 100%;"> <tr> <td colspan="2"><b>Function Enable</b></td> </tr> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This bit determines whether or not the VS stage spawns VS threads, which comprises the bulk of the VS stage functionality.</p> <p>If ENABLED, VS threads may be spawned to process VF-generated vertices before the resulting vertices are passed down the pipeline.</p> <p>If DISABLED, VF-generated vertices will pass thru the VS function and are sent down the pipeline unmodified. The Vertex Cache (if enabled) is still available.</p>	<b>Function Enable</b>		Format:	Enable					
	<b>Function Enable</b>									
Format:	Enable									
7	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">31:27</td> <td colspan="2"><b>Reserved</b></td> </tr> <tr> <td>Access:</td> <td colspan="2">RO</td> </tr> <tr> <td>Format:</td> <td colspan="2">MBZ</td> </tr> </table>	31:27	<b>Reserved</b>		Access:	RO		Format:	MBZ	
31:27	<b>Reserved</b>									
Access:	RO									
Format:	MBZ									

## 3DSTATE\_VS\_BODY

26:21	<b>Vertex URB Entry Output Read Offset</b>					
	Format:	U6				
	<p>Specifies the offset (in 256-bit units) at which Vertex URB data is to be read from the URB by the Setup Back-End (SBE) function. The offset programmed will specify the start of Attribute 0 to be passed in subsequent Pixel Shader thread payloads. Refer to the Attribute Interpolator Setup documentation.</p> <table border="1" style="width: 100%; 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;">[0,63]</td> <td></td> </tr> </tbody> </table>		Value	Name	[0,63]	
	Value	Name				
[0,63]						
<p style="text-align: center;"><b>Programming Notes</b></p> <p>As the vertex header data located at the start of the Vertex URB entry is typically only used by 3D pipeline FFs (i.e., Clipper, Setup FrontEnd) and not required as interpolated attributes in Pixel Shader threads, it is expected that SW will program this Start Offset skip over the vertex header.</p> <p>This offset value is ignored if SBE's Number of SF Attributes state is programmed to 0 (i.e., no attributes are defined beyond the position read from the Vertex Header)</p>						
20:16	<b>Vertex URB Entry Output Length</b>					
	Format:	U5				
	<p>Specifies the amount of Vertex Attribute URB data to be read by the Setup Back-End function for each Vertex URB entry, in 256-bit units. The attribute data will be read starting at the offset specified by the Vertex URB Entry Output Read Offset state.</p> <table border="1" 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;">[1,16]</td> <td></td> </tr> </tbody> </table>		Value	Name	[1,16]	
	Value	Name				
[1,16]						
<p style="text-align: center;"><b>Programming Notes</b></p> <p>This length value is ignored if SBE's Number of SF Attributes state is programmed to 0 (i.e., no attributes are defined beyond the position read from the Vertex Header).</p>						
15:8	<b>User Clip Distance Clip Test Enable Bitmask</b>					
	Format:	U8				
<p>This 8 bit mask field selects which of the 8 Clip Distance Values (if any) are to be included in the Clip stage's trivial reject / trivial accept / must clip determination function. The ClipDistance Values (if present) are located in DW8-15 of the VUE Vertex Header located at the beginning of VUE URB entries. Bit 0 of this field corresponds to Clip Distance Value 0.</p>						
7:0	<b>User Clip Distance Cull Test Enable Bitmask</b>					
	Format:	U8				
<p>This 8 bit mask field selects which of the 8 Clip Distance Values (if any) are to be included in the Clip stage's trivial reject / trivial accept determination function. Note that must clip determination is not included in this function. The ClipDistance Values (if present) are located in DW8-15 of the VUE Vertex Header located at the beginning of VUE URB entries. Bit 0 of this field corresponds to Clip Distance Value 0.</p>						

## 3DSTATE\_WM\_BODY

3DSTATE_WM_BODY															
Source:	RenderCS														
Size (in bits):	32														
Default Value:	0x00000000														
DWord	Bit	Description													
0	31	<p><b>Statistics Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>If ENABLED, the Windower and pixel pipeline will engage in statistics gathering. If DISABLED, statistics information associated with this FF stage will be left unchanged. See Statistics Gathering.</p> <table border="1"> <tr> <th colspan="2">Programming Notes</th> </tr> <tr> <td colspan="2">This bit must be disabled if any of these bits is set: 3DSTATE_WM::Legacy Depth Buffer Clear, 3DSTATE_WM::Legacy Hierarchical Depth Buffer Resolve Enable or 3DSTATE_WM::Legacy Depth Buffer Resolve Enable.</td> </tr> </table>	Format:	Enable	Programming Notes		This bit must be disabled if any of these bits is set: 3DSTATE_WM::Legacy Depth Buffer Clear, 3DSTATE_WM::Legacy Hierarchical Depth Buffer Resolve Enable or 3DSTATE_WM::Legacy Depth Buffer Resolve Enable.								
	Format:	Enable													
	Programming Notes														
	This bit must be disabled if any of these bits is set: 3DSTATE_WM::Legacy Depth Buffer Clear, 3DSTATE_WM::Legacy Hierarchical Depth Buffer Resolve Enable or 3DSTATE_WM::Legacy Depth Buffer Resolve Enable.														
	30:27	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ									
Access:	RO														
Format:	MBZ														
26	<p><b>Legacy Diamond Line Rasterization</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>This bit, if ENABLED, indicates that the Windower will rasterize zero width lines using the DX9 rasterization rules. If DISABLED, the Windower will rasterize zero width lines using the DX10 rasterization rules (see Strips Fans chapter).</p>	Format:	Enable												
Format:	Enable														
25:23	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ										
Access:	RO														
Format:	MBZ														
22:21	<p><b>Early Depth/Stencil Control</b></p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field specifies the behavior of early depth/stencil test.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>NORMAL</td> <td>Depth/Stencil Test/Write behaves as if it happens post-shader, however the pixel shader is not necessarily executed if the pixel fails depth or stencil test (this is the legacy behavior)</td> </tr> <tr> <td>1h</td> <td>PSEXEC</td> <td>Depth/Stencil Test/Write behaves as if it happens post-shader, and the pixel shader is executed if the pixel fails depth or stencil test (although pre-shader actions such as primitive inclusion, stipple, etc. will still cause the shader not to execute)</td> </tr> <tr> <td>2h</td> <td>PREPS</td> <td>Depth/Stencil Test/Write behaves as if it happens pre-shader. The pixel shader is not executed if the pixel fails depth or stencil test. Depth and stencil</td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	NORMAL	Depth/Stencil Test/Write behaves as if it happens post-shader, however the pixel shader is not necessarily executed if the pixel fails depth or stencil test (this is the legacy behavior)	1h	PSEXEC	Depth/Stencil Test/Write behaves as if it happens post-shader, and the pixel shader is executed if the pixel fails depth or stencil test (although pre-shader actions such as primitive inclusion, stipple, etc. will still cause the shader not to execute)	2h	PREPS	Depth/Stencil Test/Write behaves as if it happens pre-shader. The pixel shader is not executed if the pixel fails depth or stencil test. Depth and stencil
Format:	U2														
Value	Name	Description													
0h	NORMAL	Depth/Stencil Test/Write behaves as if it happens post-shader, however the pixel shader is not necessarily executed if the pixel fails depth or stencil test (this is the legacy behavior)													
1h	PSEXEC	Depth/Stencil Test/Write behaves as if it happens post-shader, and the pixel shader is executed if the pixel fails depth or stencil test (although pre-shader actions such as primitive inclusion, stipple, etc. will still cause the shader not to execute)													
2h	PREPS	Depth/Stencil Test/Write behaves as if it happens pre-shader. The pixel shader is not executed if the pixel fails depth or stencil test. Depth and stencil													

## 3DSTATE\_WM\_BODY

				writes occur even if the pixel is killed by the shader or post-shader by alpha test, etc. Depth output by the pixel shader is ignored.
	3h	Reserved		
<b>Programming Notes</b>				
The Early Depth/Stencil Control field cannot be set to PREPS (value = 2h) if ForceKillpix = ForceON or Forced Thread Dispatch = ForceON				
20:19	<b>Force Thread Dispatch Enable</b>			
	<b>Value</b>	<b>Name</b>	<b>Description</b>	
	0h	Normal	WM_INT::ThreadDispatchEnable is computed normally	
	1h	ForceOff	Forces WM_INT::ThreadDispatchEnable Off	
	2h	ForceON	Forces WM_INT::ThreadDispatchEnable On	
	3h	Reserved		
<b>Programming Notes</b>				
This should must always be set to Normal. This field should not be tested for functional validation				
18:17	<b>Position ZW Interpolation Mode</b>			
	Format:		U2	
This field elects "interpolation mode" associated with the Position Z (source depth) and W coordinates passed in the PS payload when the PS requires Position as input. This field does not determine whether these coordinates are actually included in the payload (see Pixel Shader Requires Depth, Pixel Shader Requires W).				
	<b>Value</b>	<b>Name</b>	<b>Description</b>	
	0h	INTERP_PIXEL	Evaluate Z & W at the pixel center or UL corner (as specified by Pixel Location of 3DSTATE_MULTISAMPLE)	
	1h	Reserved		
	2h	INTERP_CENTROID		
	3h	INTERP_SAMPLE		
<b>Programming Notes</b>				
WM_INT::RT Independent Rasterization Enable must be disabled in order to select INTERP_SAMPLE.				
MSDISPMODE_PERSAMPLE is required in order to select INTERP_SAMPLE.				
16:11	<b>Barycentric Interpolation Mode</b>			
	Format:		Enable[6]	
Controls which barycentric interpolation terms must be passed into the pixel shader kernel.				
Bit 0: Perspective Pixel Location barycentric is required				
Bit 1: Perspective Centroid barycentric is required				
Bit 2: Perspective Sample barycentric is required				

## 3DSTATE\_WM\_BODY

		Bit 3: Non-perspective Pixel Location barycentric is required Bit 4: Non-perspective Centroid barycentric is required Bit 5: Non-perspective Sample barycentric is required	
		<b>Programming Notes</b>	
		If contiguous dispatch modes are enabled, only bit 3 (non-perspective pixel location) can be set, all other bits in this field must be zero. Pixel Location below refers to either the upper left corner or pixel center depending on the <b>Pixel Location</b> state of 3DSTATE_MULTISAMPLING).MSDISPMODE_PERSAMPLE is required in order to select Perspective Sample or Non-perspective Sample barycentric coordinates.	
10	<b>Reserved</b>	Access:	RO
		Format:	MBZ
9:8	<b>Line End Cap Antialiasing Region Width</b>	Format:	U2
		This field specifies the distances over which the coverage of anti-aliased line end caps are computed.	
		<b>Value</b>	<b>Name</b>
		0h	0.5 pixels
		1h	1.0 pixels
		2h	2.0 pixels
		3h	4.0 pixels
7:6	<b>Line Antialiasing Region Width</b>	Format:	U2
		This field specifies the distance over which the anti-aliased line coverage is computed.	
		<b>Value</b>	<b>Name</b>
		0h	0.5 pixels
		1h	1.0 pixels
		2h	2.0 pixels
		3h	4.0 pixels
5	<b>Reserved</b>	Access:	RO
		Format:	MBZ
4	<b>Polygon Stipple Enable</b>	Format:	Enable
		Enables the Polygon Stipple function.	
3	<b>Line Stipple Enable</b>	Format:	Enable
		Enables the Line Stipple function.	



<b>3DSTATE_WM_BODY</b>		
2	<b>Point Rasterization Rule</b> This field specifies the rasterization rules to be applied whenever the edges of a point primitive fall exactly on a pixel sampling point.	
	<b>Value</b>	<b>Name</b>
	<b>Description</b>	
0h	RASTRULE_UPPER_LEFT	To match "normal" upper left rules for surface primitives
1h	RASTRULE_UPPER_RIGHT	To match OpenGL point rasterization rules (round to + infinity, where this is the upper right direction wrt OpenGL screen origin of lower left).
1:0	<b>Force Kill Pixel Enable</b>	
	<b>Value</b>	<b>Name</b>
	<b>Description</b>	
0h	Normal	WM_INT:: Pixel Shader Kill Pixel is computed normally
1h	ForceOff	Forces WM_INT:: Pixel Shader Kill Pixel Off
2h	ForceON	Forces WM_INT:: Pixel Shader Kill Pixel On
3h	Reserved	
	<b>Programming Notes</b>	
	This should must always be set to Normal. This field should not be tested for functional validation	



## 3DSTATE\_WM\_CHROMAKEY\_BODY

3DSTATE_WM_CHROMAKEY_BODY		
Source:	RenderCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31	<b>ChromaKey Kill Enable</b> Format: Enable If ENABLED, indicates that at least one of the attached samplers has ChromaKeyKill enabled.
	30:0	<b>Reserved</b> Access: RO Format: MBZ

## 3DSTATE\_WM\_DEPTH\_STENCIL\_BODY

3DSTATE_WM_DEPTH_STENCIL_BODY								
Source:	RenderCS							
Size (in bits):	96							
Default Value:	0x00000000, 0x00000000, 0x00000000							
DWord	Bit	Description						
0	31:29	<p><b>Stencil Fail Op</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Stencil_Operation</b></td> </tr> </table> <p>This field specifies the operation to perform on the Stencil Buffer when the (front face) stencil test fails.</p> <table border="1"> <tr> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="2">if all three stencil ops (Stencil Fail, Stencil Pass Depth Fail, and Stencil Pass Depth Pass) are KEEP, ZERO, or REPLACE, the stencil buffer is not read.</td> </tr> </table>	Format:	<b>3D_Stencil_Operation</b>	<b>Programming Notes</b>		if all three stencil ops (Stencil Fail, Stencil Pass Depth Fail, and Stencil Pass Depth Pass) are KEEP, ZERO, or REPLACE, the stencil buffer is not read.	
	Format:	<b>3D_Stencil_Operation</b>						
	<b>Programming Notes</b>							
	if all three stencil ops (Stencil Fail, Stencil Pass Depth Fail, and Stencil Pass Depth Pass) are KEEP, ZERO, or REPLACE, the stencil buffer is not read.							
	28:26	<p><b>Stencil Pass Depth Fail Op</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Stencil_Operation</b></td> </tr> </table> <p>This field specifies the operation to perform on the Stencil Buffer when the (front face) stencil test passes but the depth pass fails.</p>	Format:	<b>3D_Stencil_Operation</b>				
	Format:	<b>3D_Stencil_Operation</b>						
	25:23	<p><b>Stencil Pass Depth Pass Op</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Stencil_Operation</b></td> </tr> </table> <p>This field specifies the operation to perform on the Stencil Buffer when the (front face) stencil test passes but the depth test passes.</p>	Format:	<b>3D_Stencil_Operation</b>				
	Format:	<b>3D_Stencil_Operation</b>						
22:20	<p><b>Backface Stencil Test Function</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Compare_Function</b></td> </tr> </table>	Format:	<b>3D_Compare_Function</b>					
Format:	<b>3D_Compare_Function</b>							
19:17	<p><b>Backface Stencil Fail Op</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Stencil_Operation</b></td> </tr> </table>	Format:	<b>3D_Stencil_Operation</b>					
Format:	<b>3D_Stencil_Operation</b>							
16:14	<p><b>Backface Stencil Pass Depth Fail Op</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Stencil_Operation</b></td> </tr> </table> <p>This field specifies the operation to perform on the Stencil Buffer when the stencil test passes but the depth pass fails.</p>	Format:	<b>3D_Stencil_Operation</b>					
Format:	<b>3D_Stencil_Operation</b>							
13:11	<p><b>Backface Stencil Pass Depth Pass Op</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Stencil_Operation</b></td> </tr> </table> <p>This field specifies the operation to perform on the Stencil Buffer when the stencil test passes and the depth pass passes (or is disabled).</p>	Format:	<b>3D_Stencil_Operation</b>					
Format:	<b>3D_Stencil_Operation</b>							
10:8	<p><b>Stencil Test Function</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Compare_Function</b></td> </tr> </table> <p>This field specifies the comparison function used in the (front face) StencilTest function.</p>	Format:	<b>3D_Compare_Function</b>					
Format:	<b>3D_Compare_Function</b>							

## 3DSTATE\_WM\_DEPTH\_STENCIL\_BODY

7:5	<b>Depth Test Function</b>	Format:	<b>3D_Compare_Function</b>
Specifies the comparison function used in DepthTest function.			
<b>Programming Notes</b>			
If the Depth Test Function is ALWAYS or NEVER, the depth buffer is not read.			
4	<b>Double Sided Stencil Enable</b>	Format:	Enable
Enable doubled sided stencil operations.			
<b>Value</b>		<b>Name</b>	<b>Description</b>
0h		False	Double Sided Stencil Disabled
1h		True	Double Sided Stencil Enabled
<b>Programming Notes</b>			
<ul style="list-style-type: none"> <li>Back-facing primitives have a vertex winding order opposite to the currently selected Front Winding state.</li> <li>Culling of primitives is not affected by the double sided stencil state</li> <li>Back-facing primitives will be rendered, honoring all current device state, as though it were a front-facing primitive with no implicitly overloaded state.</li> </ul>			
3	<b>Stencil Test Enable</b>	Format:	Enable
Enables StencilTest function of the Pixel Processing pipeline.			
<b>Programming Notes</b>			
If any of the render targets are YUV format, this field must be disabled.			
2	<b>Stencil Buffer Write Enable</b>	Format:	Enable
Enables writes to the Stencil Buffer.			
<b>Programming Notes</b>			
If this field is enabled, Stencil Test Enable must also be enabled.			
1	<b>Depth Test Enable</b>	Format:	Enable
Enables the DepthTest function of the Pixel Processing pipeline.			
<b>Value</b>		<b>Name</b>	
0h		Disable	
1h		Enable	
<b>Programming Notes</b>			
If any of the render targets are YUV format, this field must be disabled.			

<b>3DSTATE_WM_DEPTH_STENCIL_BODY</b>	
0	<b>Depth Buffer Write Enable</b>
	Format: <span style="float: right;">Enable</span>
	Enables writes to the Depth Buffer.
	<b>Programming Notes</b>
	A Depth Buffer must be defined before enabling writes to it, or operation is UNDEFINED. This bit must not be set when WM_INT::RT Independent Rasterization Enable is true.
1	31:24 <b>Stencil Test Mask</b>
	Format: <span style="float: right;">U8</span>
	This field specifies a bit mask applied to stencil test values. Both the stencil reference value and value read from the stencil buffer will be logically ANDed with this mask before the stencil comparison test is performed.
	23:16 <b>Stencil Write Mask</b>
Format: <span style="float: right;">U8</span>	
This field specifies a bit mask applied to stencil buffer writes. Only those stencil buffer bits corresponding to bits set in this mask will be modified.	
15:8 <b>Backface Stencil Test Mask</b>	
Format: <span style="float: right;">U8</span>	
This field specifies a bit mask applied to backface stencil test values. Both the stencil reference value and value read from the stencil buffer will be logically ANDed with this mask before the stencil comparison test is performed.	
7:0 <b>Backface Stencil Write Mask</b>	
Format: <span style="float: right;">U8</span>	
This field specifies a bit mask applied to backface stencil buffer writes. Only those stencil buffer bits corresponding to bits set in this mask will be modified.	
2	31:16 <b>Reserved</b>
	Access: <span style="float: right;">RO</span>
	Format: <span style="float: right;">MBZ</span>
	15:8 <b>Stencil Reference Value</b>
Format: <span style="float: right;">U8</span>	
This field specifies the stencil reference value to compare against in the (front face) StencilTest function.	
7:0 <b>Backface Stencil Reference Value</b>	
Format: <span style="float: right;">U8</span>	
This field specifies the stencil reference value to compare against in the StencilTest function.	

## 3DSTATE\_WM\_HZ\_OP\_BODY

3DSTATE_WM_HZ_OP_BODY				
Source:	RenderCS			
Size (in bits):	128			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31	<p><b>Stencil Buffer Clear Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>When set, the stencil buffer is initialized.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>If this field is enabled,</p> <ol style="list-style-type: none"> <li>the <b>Depth Buffer Resolve Enable (full or partial)</b> and <b>Hierarchical Depth Buffer Resolve Enable</b> fields must both be disabled.</li> <li>3DSTATE_DEPTH_BUFFER::Stencil Write Enable must be set if 3DSTATE_STENCIL_BUFFER::Stencil buffer enable is set.</li> </ol> <p>When this field is enabled, Stencil Buffer Resolve Enable should be disabled</p>	Format:	Enable
	Format:	Enable		
	30	<p><b>Depth Buffer Clear Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>When set, the depth buffer is initialized.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>If this field is enabled,</p> <ol style="list-style-type: none"> <li>the <b>Depth Buffer Resolve Enable</b>(full or partial) and <b>Hierarchical Depth Buffer Resolve Enable</b> fields must both be disabled.</li> <li>3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set.</li> </ol>	Format:	Enable
Format:	Enable			
29	<p><b>Scissor Rectangle Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>Enables operation of Scissor Rectangle.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>In order get the functionality right if this bit is disabled, driver must clip the clear rectangle to scissor rectangle if scissor test is enabled before clearing.</p>	Format:	Enable	
Format:	Enable			

## 3DSTATE\_WM\_HZ\_OP\_BODY

	28	<b>Depth Buffer Resolve Enable</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>When set, the depth buffer is made to be consistent with the hierarchical depth buffer as a side-effect of rendering pixels. This is intended to be used when the depth buffer is to be used as a surface outside of the 3D rendering operation. The Depth buffer will be in uncompressed state after this operation.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> <tr> <td> <p>If this field is enabled,</p> <ol style="list-style-type: none"> <li>1. the <b>Depth Buffer Clear</b> and <b>Hierarchical Depth Buffer Resolve Enable</b> fields must both be disabled.</li> <li>2. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set.</li> <li>3. Depth Buffer Partial Resolve Enable field should be disabled.</li> </ol> <p>For validation reasons, the need to resolve an area smaller than the whole depth buffer can occur. See the programming notes for X/Y Min and X/Y Max</p> </td> </tr> </table>	Format:	Enable	Programming Notes	<p>If this field is enabled,</p> <ol style="list-style-type: none"> <li>1. the <b>Depth Buffer Clear</b> and <b>Hierarchical Depth Buffer Resolve Enable</b> fields must both be disabled.</li> <li>2. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set.</li> <li>3. Depth Buffer Partial Resolve Enable field should be disabled.</li> </ol> <p>For validation reasons, the need to resolve an area smaller than the whole depth buffer can occur. See the programming notes for X/Y Min and X/Y Max</p>
Format:	Enable						
Programming Notes							
<p>If this field is enabled,</p> <ol style="list-style-type: none"> <li>1. the <b>Depth Buffer Clear</b> and <b>Hierarchical Depth Buffer Resolve Enable</b> fields must both be disabled.</li> <li>2. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set.</li> <li>3. Depth Buffer Partial Resolve Enable field should be disabled.</li> </ol> <p>For validation reasons, the need to resolve an area smaller than the whole depth buffer can occur. See the programming notes for X/Y Min and X/Y Max</p>							
	27	<b>Hierarchical Depth Buffer Resolve Enable</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <p>When set, the hierarchical depth buffer is made to be consistent with the depth buffer as a side-effect of rendering pixels. This is intended to be used when the depth buffer has been modified outside of the 3D rendering operation.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; color: #0070C0;">Programming Notes</th> </tr> <tr> <td> <p>If this field is enabled,</p> <ol style="list-style-type: none"> <li>1. the <b>Depth Buffer Clear</b> and <b>Depth Buffer Resolve Enable (full or partial)</b> fields must both be disabled.</li> <li>2. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set.</li> <li>3. Stencil Buffer Resolve Enable must be disabled.</li> </ol> <p>Doing a Hierarchical Depth Buffer resolve (HZ resolve) on a partial HZ buffer is not permitted. The HZ resolve operation must be done on the entire HZ buffer.</p> <p><b>Performance Note:</b> expect the hierarchical depth buffer's impact on performance to be reduced for some period of time after this operation is performed, as the hierarchical depth buffer is initialized to a state that makes it ineffective. Further rendering will tend to bring the hierarchical depth buffer back to a more effective state.</p> </td> </tr> </table>	Format:	Enable	Programming Notes	<p>If this field is enabled,</p> <ol style="list-style-type: none"> <li>1. the <b>Depth Buffer Clear</b> and <b>Depth Buffer Resolve Enable (full or partial)</b> fields must both be disabled.</li> <li>2. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set.</li> <li>3. Stencil Buffer Resolve Enable must be disabled.</li> </ol> <p>Doing a Hierarchical Depth Buffer resolve (HZ resolve) on a partial HZ buffer is not permitted. The HZ resolve operation must be done on the entire HZ buffer.</p> <p><b>Performance Note:</b> expect the hierarchical depth buffer's impact on performance to be reduced for some period of time after this operation is performed, as the hierarchical depth buffer is initialized to a state that makes it ineffective. Further rendering will tend to bring the hierarchical depth buffer back to a more effective state.</p>
Format:	Enable						
Programming Notes							
<p>If this field is enabled,</p> <ol style="list-style-type: none"> <li>1. the <b>Depth Buffer Clear</b> and <b>Depth Buffer Resolve Enable (full or partial)</b> fields must both be disabled.</li> <li>2. 3DSTATE_DEPTH_BUFFER::Depth Write Enable must be set.</li> <li>3. Stencil Buffer Resolve Enable must be disabled.</li> </ol> <p>Doing a Hierarchical Depth Buffer resolve (HZ resolve) on a partial HZ buffer is not permitted. The HZ resolve operation must be done on the entire HZ buffer.</p> <p><b>Performance Note:</b> expect the hierarchical depth buffer's impact on performance to be reduced for some period of time after this operation is performed, as the hierarchical depth buffer is initialized to a state that makes it ineffective. Further rendering will tend to bring the hierarchical depth buffer back to a more effective state.</p>							

## 3DSTATE\_WM\_HZ\_OP\_BODY

	26	<p><b>Pixel Position Offset Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>Enables the device to offset pixel positions by 0.5 both in horizontal and vertical directions.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>Setting this field along with setting the Pixel Location to upper left and number of multisamples to greater than one will cause the device to offset pixel positions by 0.5 both in horizontal and vertical directions. It is to be noted this is done to adjust the pixel co-ordinate system to DX9 like, so any WM_HZ_OP screen space rectangles (eg: legacy HiZ Clear, Resolve etc) generated internally by driver in this mode needs to be aware of this offset adjustment and send the rectangles according to alignment restriction taking this offset adjustment into consideration. SW can choose to set this bit only for DX9 API. DX10/OpenGL API's should not have any effect by setting or not setting this bit.</p>	Format:	Enable	<b>Programming Notes</b>
	Format:	Enable			
	<b>Programming Notes</b>				
25	<p><b>Full Surface Depth and Stencil Clear</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>Setting this field to "1" along with "Depth buffer clear" will cause all the pixels/samples in an the HZ and Stencil CLs to be cleared. Software must set this only when the APP requires the entire Depth surface to be cleared. Setting this field to "1" for STC-buffer only clear without "depth buffer clear" will cause all the pixels/samples in the STC-CL to get the stc-ref value.</p>	Format:	Enable	<b>Programming Notes</b>	
Format:	Enable				
<b>Programming Notes</b>					
24	<p><b>Stencil Buffer Resolve Enable</b></p> <p>When set, the stencil buffer is filled with the true stencil values. This is intended to be used when the stencil buffer is to be used as a surface outside of the 3D rendering operation. When this is enabled, Stencil Buffer Clear Enable field should be disabled.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>The STC buffer is required to be done only if the "compression bit" in the 3DSTATE_STENCIL_BUFFER is enabled. If STC buffer compression is enabled, then the STC buffer will hold compressed data. To get the true stencil values for all the pixels, a stencil resolve operation is required.</p> <p><u>Issue</u></p> <p>The stencil resolve bit in WM_HZ_OP state is not being</p>	<b>Programming Notes</b>			
<b>Programming Notes</b>					



## 3DSTATE\_WM\_HZ\_OP\_BODY

		<p>considered by WM in the right pipeline stage</p> <p><u>Workaround</u></p> <p><b>Astep only</b> W/a is to have a pipe control with thread_dispatch set to OFF before the WM_HZ_OP with stencil buffer resolve is required(bit 24 below).</p> <p>1) Setting the force thread dispatch enable(bits 20:19) in the 3dstate_WM_body state to be set to Force_OFF (value of 1) before the first WM_HZ_OP state cycle</p> <p>2) Before second WM_HZ_OP state which is required by programming sequencing to complete the HZ_OP operation, reprogram the 3dstate_WM_body to set to NORMAL(value of 0).</p> <p>3DSTATE_WM.ForceThreadDispatchEnable = 1(ForceOff)</p> <p>PIPE_CONTROL commit the above state before HZ_OP</p> <p>3DSTATE_WM_HZ_OP (stencil resolve bit set)</p> <p>PIPE_CONTROL</p> <p>3DSTATE_WM.ForceThreadDispatchEnable = 0(Normal)</p> <p>3DSTATE_WM_HZ_OP (empty, no bits set)</p>																								
	23:16	<p><b>Stencil Clear Value</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8.0</td> </tr> </table> <p>This field specifies the stencil clear value.</p>		Format:	U8.0																					
Format:	U8.0																									
	15:13	<p><b>Number of Multisamples</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U3</td> </tr> </table> <p>This field specifies how many samples/pixel exist in the Depth Buffer and Stencil buffers, as <math>\log_2(\#samples)</math>.</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 20%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>1</td> <td>1 sample/pixel</td> </tr> <tr> <td>1h</td> <td>2</td> <td>2 samples/pixel</td> </tr> <tr> <td>2h</td> <td>4</td> <td>4 samples/pixel</td> </tr> <tr> <td>3h</td> <td>8</td> <td>8 samples/pixel</td> </tr> <tr> <td>4h</td> <td>16</td> <td>16 samples/pixel</td> </tr> <tr> <td>5h-7h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>		Format:	U3	Value	Name	Description	0h	1	1 sample/pixel	1h	2	2 samples/pixel	2h	4	4 samples/pixel	3h	8	8 samples/pixel	4h	16	16 samples/pixel	5h-7h	Reserved	
Format:	U3																									
Value	Name	Description																								
0h	1	1 sample/pixel																								
1h	2	2 samples/pixel																								
2h	4	4 samples/pixel																								
3h	8	8 samples/pixel																								
4h	16	16 samples/pixel																								
5h-7h	Reserved																									
	12:9	<p><b>Reserved</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ																			
Access:	RO																									
Format:	MBZ																									
	8:0	<p><b>Reserved</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ																			
Access:	RO																									
Format:	MBZ																									

<b>3DSTATE_WM_HZ_OP_BODY</b>				
<p style="text-align: center;">1</p> <p><b>Programming Notes:</b></p> <p>The Clear/Resolve rectangle X and Y Min values must be shifted by the LOD level; i.e. the hardware does not include the LOD in this function. Hence to clear any particular X, Y from the base level, to clear the contents at level "LOD" use (X»LOD) and (Y»LOD).</p> <p>The final X and Y Min values, after LOD adjustment described above, have to be manually 8x4 or 8x8 aligned for Depth and HZ Resolve passes only. For Clears see "Full Surface Depth and Stencil Clear" field in this command instead.</p> <p>For multisample set to 1X together with Z format set to 16 bpp:</p> <pre> resolve_aligned_y_min = (y_min &amp; ~0x7) //round down to last multiple of 8 resolve_aligned_x_min = (x_min &amp; ~0x7) //round down to last multiple of 8  All other cases: resolve_aligned_y_min = (y_min &amp; ~0x3) //round down to last multiple of 4 resolve_aligned_x_min = (x_min &amp; ~0x7) //round down to last multiple of 8 </pre>	31:16	<p><b>Clear Rectangle Y Min</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>Specifies Ymin value of (inclusive) of clear rectangle with the Depth Buffer, used for clipping. Pixels with Y coordinates less than Ymin will not be affected.</p>	Format:	U16
	Format:	U16		
15:0	<p><b>Clear Rectangle X Min</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>Specifies Xmin value of (inclusive) of clear rectangle with the Depth Buffer, used for clipping. Pixels with X coordinates less than or equal to Xmin will not be affected.</p>	Format:	U16	
Format:	U16			
<p style="text-align: center;">2</p> <p><b>Programming Notes:</b></p> <p>See the programming note in the previous DWORD for the Min values. The Clear/Resolve rectangle X and YMax values must be shifted by the LOD level; i.e. the hardware does not include the LOD in this function. Hence to clear any particular X, Y from the base level, to clear the contents at level "LOD" use (X»LOD) and (Y»LOD). The final X and Y Max values, after LOD adjustment described above, have to be manually 8x4 or 8x8 aligned for Depth and HZ Resolve passes only. For</p>	31:16	<p><b>Clear Rectangle Y Max</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>Specifies Ymax value of (exclusive) of clear rectangle with the Depth Buffer, used for clipping. Pixels with Y coordinates greater than Ymax will be not be cleared.</p>	Format:	U16
	Format:	U16		
15:0	<p><b>Clear Rectangle X Max</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>Specifies Xmax value of (exclusive) of clear rectangle with the Depth Buffer, used for clipping. Pixels with X coordinates greater than or equal to Xmax will be not be affected.</p>	Format:	U16	
Format:	U16			

<b>3DSTATE_WM_HZ_OP_BODY</b>									
<p>Clears see "Full Surface Depth and Stencil Clear" field in this command instead.</p> <p>For multisample set to 1X together with Z format set to 16 bpp:</p> <pre> resolve_aligned_y_max= (y_max &amp; ~0x7) + ((y_max &amp; 0x7 == 0) ? 0 : 8) //round up to next multiple of 8 resolve_aligned_x_max= (x_max &amp; ~0x7) + ((x_max &amp; 0x7 == 0) ? 0 : 8) //round up to next multiple of 8 All other cases: resolve_aligned_y_max= (y_max &amp; ~0x3) + ((y_max &amp; 0x3 == 0) ? 0 : 4) //round up to next multiple of 4 resolve_aligned_x_max= (x_max &amp; ~0x7) + ((x_max &amp; 0x7 == 0) ? 0 : 8) //round up to next multiple of 8 </pre>									
3	<p>31:16 <b>Reserved</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Access:</td> <td style="padding: 2px;">RO</td> </tr> <tr> <td style="padding: 2px;">Format:</td> <td style="padding: 2px;">MBZ</td> </tr> </table> <p>15:0 <b>Sample Mask</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">           Format:            Right-justified bitmask (Bit 0 = Sample0). Number of bits that are used is determined by Num Multisamples (3DSTATE_WM_HZ_OP)         </td> </tr> <tr> <td style="padding: 2px;">           A per-multisample-position mask state variable that is immediately and unconditionally ANDed with the sample coverage mask as part of the rasterization process. This mask is applied prior to centroid selection.         </td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; padding: 2px;">Programming Notes</th> </tr> <tr> <td style="padding: 2px;">           If Number of Multisamples is NUMSAMPLES_1, bits 15:1 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_2, bits 15:2 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_4, bits 15:4 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_8, bits 15:8 of this field will be zeroed by HW.         </td> </tr> </table>	Access:	RO	Format:	MBZ	Format: Right-justified bitmask (Bit 0 = Sample0). Number of bits that are used is determined by Num Multisamples (3DSTATE_WM_HZ_OP)	A per-multisample-position mask state variable that is immediately and unconditionally ANDed with the sample coverage mask as part of the rasterization process. This mask is applied prior to centroid selection.	Programming Notes	If Number of Multisamples is NUMSAMPLES_1, bits 15:1 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_2, bits 15:2 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_4, bits 15:4 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_8, bits 15:8 of this field will be zeroed by HW.
Access:	RO								
Format:	MBZ								
Format: Right-justified bitmask (Bit 0 = Sample0). Number of bits that are used is determined by Num Multisamples (3DSTATE_WM_HZ_OP)									
A per-multisample-position mask state variable that is immediately and unconditionally ANDed with the sample coverage mask as part of the rasterization process. This mask is applied prior to centroid selection.									
Programming Notes									
If Number of Multisamples is NUMSAMPLES_1, bits 15:1 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_2, bits 15:2 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_4, bits 15:4 of this field will be zeroed by HW.If Number of Multisamples is NUMSAMPLES_8, bits 15:8 of this field will be zeroed by HW.									



## A32 Buffer Base Address Message Header Control

<b>MHC_A32_BBA - A32 Buffer Base Address Message Header Control</b>								
Size (in bits):		32						
Default Value:		0x00000000						
<b>DWord</b>	<b>Bit</b>	<b>Description</b>						
0	31:10	<b>Buffer Base Address Offset</b>						
		<table border="1"> <tr> <td>Format:</td> <td>GeneralStateOffset[31:10]</td> </tr> <tr> <td colspan="2">Specifies the base address offset page [31:10] for A32 stateless messages.</td> </tr> <tr> <td colspan="2" style="text-align: center;"><b>Restriction</b></td> </tr> <tr> <td colspan="2">When using stateless A32 Data Port messages, General State Base Address[47:12] + Buffer Base Address[31:10] must be less than <math>2^{48}</math>. It is illegal for this to be greater or equal than <math>2^{48}</math>.</td> </tr> </table>	Format:	GeneralStateOffset[31:10]	Specifies the base address offset page [31:10] for A32 stateless messages.		<b>Restriction</b>	
Format:	GeneralStateOffset[31:10]							
Specifies the base address offset page [31:10] for A32 stateless messages.								
<b>Restriction</b>								
When using stateless A32 Data Port messages, General State Base Address[47:12] + Buffer Base Address[31:10] must be less than $2^{48}$ . It is illegal for this to be greater or equal than $2^{48}$ .								
	9:0	<b>Reserved</b>						
		Access:	RO					
		Format:	MBZ					

## A32 Scaled Header Present Message Descriptor Control Field

MDC_A32_MHP - A32 Scaled Header Present Message Descriptor Control Field																	
Size (in bits):		1															
Default Value:		0x00000000															
DWord	Bit	Description															
0	0	<p><b>Message Header Present</b></p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">Boolean</td> </tr> <tr> <td colspan="3">Specifies if the message uses the optional message header to modify the A32 address calculation, in combination with the MDC_A32_SSO field.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>0h</td> <td>No <b>[Default]</b></td> <td>Message header is not present.</td> </tr> <tr> <td>1h</td> <td>Yes</td> <td>Message header is present.</td> </tr> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>The access is Out-of-bounds if the SideB and Offset is enabled when the Message Header is not present.</p>	Format:	Boolean		Specifies if the message uses the optional message header to modify the A32 address calculation, in combination with the MDC_A32_SSO field.			Value	Name	Description	0h	No <b>[Default]</b>	Message header is not present.	1h	Yes	Message header is present.
Format:	Boolean																
Specifies if the message uses the optional message header to modify the A32 address calculation, in combination with the MDC_A32_SSO field.																	
Value	Name	Description															
0h	No <b>[Default]</b>	Message header is not present.															
1h	Yes	Message header is present.															



## A32 Sideband Scale and Offset Enable Message Descriptor Control Field

MDC_A32_SBSO - A32 Sideband Scale and Offset Enable Message Descriptor Control Field				
Size (in bits):		8		
Default Value:		0x00000000		
DWord	Bit	Description		
0	7	<p><b>Sideband Offset Enable</b></p> <table border="1"> <tr> <td>Format:</td> <td>MBO</td> </tr> </table> <p>Must be set for a scaled SLM access. The 16-bit offset from the Sideband is added to all the offsets in the Address Payload for the SLM access. The 16-bit Sideband Offset is specified in the extended function control field in the SEND instruction.</p>	Format:	MBO
	Format:	MBO		
6:0	<p><b>Scale</b></p> <table border="1"> <tr> <td>Format:</td> <td>U7</td> </tr> </table> <p>Specifies the scale pitch to be used for SLM messages as (#bytes-1).</p>	Format:	U7	
Format:	U7			

## A64 Data Size Message Descriptor Control Field

MDC_A64_DS - A64 Data Size Message Descriptor Control Field														
Size (in bits):		2												
Default Value:		0x00000000												
DWord	Bit	Description												
0	1:0	<p><b>Data Size</b> Specifies the number of data elements to be read or written</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>DE1</td> <td>1 data element (B, DW, QW)</td> </tr> <tr> <td>01h</td> <td>DE2</td> <td>2 data elements (B, DW, QW)</td> </tr> <tr> <td>02h</td> <td>DE4</td> <td>4 data elements (B, DW, QW)</td> </tr> </tbody> </table> <p><b>Restriction</b></p> <p>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.</p>	Value	Name	Description	00h	DE1	1 data element (B, DW, QW)	01h	DE2	2 data elements (B, DW, QW)	02h	DE4	4 data elements (B, DW, QW)
Value	Name	Description												
00h	DE1	1 data element (B, DW, QW)												
01h	DE2	2 data elements (B, DW, QW)												
02h	DE4	4 data elements (B, DW, QW)												



## A64 Hword Block Message Header

<b>MH_A64_HWB - A64 Hword Block Message Header</b>		
Source:	EuSubFunctionDataPort1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:0	<b>BlockOffset</b>
		Format: U64
		Specifies the U64 byte offset of Oword block.
		<b>Programming Notes</b>
		If the BlockOffset is not in the 48-bit canonical address range, the access is Out-of-Bounds.
		<b>Restriction</b>
		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	<b>Reserved</b>
		Access: RO
		Format: MBZ
5	31:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
6..7	63:0	<b>Reserved</b>
		Access: RO
		Format: MBZ



## A64 Hword Data Blocks Message Descriptor Control Field

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

<b>MH_A64_OWB - A64 Oword Block Message Header</b>		
Source:	EuSubFunctionDataPort1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:0	<b>BlockOffset</b>
		Format: U64
		Specifies the U64 byte offset of Oword block.
		<b>Programming Notes</b>
		If the BlockOffset is not in the 48-bit canonical address range, the access is Out-of-Bounds.
		<b>Restriction</b>
		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	<b>Reserved</b>
		Access: RO
		Format: MBZ

## A64 Oword Data Blocks Message Descriptor Control Field

MDC_A64_DB_OW - A64 Oword Data Blocks Message Descriptor Control Field																							
Size (in bits):		3																					
Default Value:		0x00000000																					
DWord	Bit	Description																					
0	2:0	<p><b>Data Blocks</b> Specifies the number of Oword blocks to be read or written</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>OW1L</td> <td>1 Oword, read into or written from the low 128 bits of the destination register</td> </tr> <tr> <td>01h</td> <td>Reserved</td> <td>Reserved</td> </tr> <tr> <td>02h</td> <td>OW2</td> <td>2 Owords</td> </tr> <tr> <td>03h</td> <td>OW4</td> <td>4 Owords</td> </tr> <tr> <td>04h</td> <td>OW8</td> <td>8 Owords</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Value	Name	Description	00h	OW1L	1 Oword, read into or written from the low 128 bits of the destination register	01h	Reserved	Reserved	02h	OW2	2 Owords	03h	OW4	4 Owords	04h	OW8	8 Owords	Others	Reserved	Ignored
Value	Name	Description																					
00h	OW1L	1 Oword, read into or written from the low 128 bits of the destination register																					
01h	Reserved	Reserved																					
02h	OW2	2 Owords																					
03h	OW4	4 Owords																					
04h	OW8	8 Owords																					
Others	Reserved	Ignored																					



## A64 Scaled Header Present Message Descriptor Control Field

MDC_A64_MHP - A64 Scaled Header Present Message Descriptor Control Field											
Size (in bits):		1									
Default Value:		0x00000000									
DWord	Bit	Description									
0	0	<p><b>Message Header Present</b> 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"> <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> <p style="text-align: center;"><b>Programming Notes</b></p> <p>The access is Out-of-Bounds if the SideBand Offset is enabled when the Message Header is not present.</p>	Value	Name	Description	0h	No	Message header is not present	1h	Yes	Message header is present
Value	Name	Description									
0h	No	Message header is not present									
1h	Yes	Message header is present									

## AddrSubRegNum

<b>AddrSubRegNum</b>						
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	<b>Address Subregister Number</b> <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;">0-15</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																							
Size (in bits):	8																						
Default Value:	0x00000000																						
DWord	Bit	Description																					
0	7:0	<p><b>Binding Table Index</b> Specifies the surface for the message, which can be Surface State Model, SLM or Stateless.</p> <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>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> </tbody> </table> <p style="text-align: center;"><b>Restriction</b></p> <p>When using A32_A64_NC, SW must ensure that 2 threads do not both access the same cache line (64B)</p>	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	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).
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																					
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

<b>MDC_FOP2 - Atomic Float Binary Operation Message Descriptor Control Field</b>					
Size (in bits):		3			
Default Value:		0x00000001			
DWord	Bit	Description			
0	2:0	<b>Atomic Float Operation Type</b> Specifies the atomic float binary operation to be performed			
		Value	Name		
		Description	Programming Notes		
		01h	<b>AOP_FMAX</b> <b>[Default]</b>	new_dst = fmax(old_dst, src0)	The fmax operation implements the IEEE specification, which differs slightly from the DX and OCL specifications 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(-0, +0) = fmax(+0, -0) = +0 Fmax with sNaN operand returns sNaN instead of quietize(sNaN)
		02h	<b>AOP_FMIN</b>	new_dst = fmin(old_dst, src0)	The fmin operation implements the IEEE specification, which differs slightly from the DX and OCL specifications 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(+0, -0) = fmin(-0, +0) = -0 Fmin with sNaN operand returns sNaN instead of quietize(sNaN) [] Fmin(+0,-0) returns +0. Should be -0, to match EU Fmin instruction.

## Atomic Float Operation Message Descriptor Control Field

<b>MDC_FOP - Atomic Float Operation Message Descriptor Control Field</b>																							
Size (in bits):		3																					
Default Value:		0x00000000																					
DWord	Bit	Description																					
0	2:0	<b>Atomic Float Operation Type</b> Specifies the atomic float operation to be performed.																					
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>01h</td> <td>AOP2_FMAX</td> <td>new_dst = fmax(old_dst, src0) (default binary opcode)</td> <td>The fmax operation implements the IEEE specification, which differs slightly from the DX and OCL specifications when a source operand is a sNaN.  <math>fmax(x, qNaN) = fmax(qNaN, x) = x</math>  <math>fmax(x, sNaN) = fmax(sNaN, x) = quietize(sNaN)</math>  <math>fmax(sNaN, sNaN) = fmax(sNaN, qNaN) = fmax(qNaN, sNaN) = quietize(sNaN)</math>  <math>fmax(qNaN, qNaN) = qNaN</math>  <math>fmax(-0, +0) = fmax(+0, -0) = +0</math>            Fmax with sNaN operand returns sNaN instead of quietize(sNaN)         </td> </tr> <tr> <td>02h</td> <td>AOP2_FMIN</td> <td>new_dst = fmin(old_dst, src0)</td> <td>The fmin operation implements the IEEE specification, which differs slightly from the DX and OCL specifications when a source operand is a sNaN.  <math>fmin(x, qNaN) = fmin(qNaN, x) = x</math>  <math>fmin(x, sNaN) = fmin(sNaN, x) = quietize(sNaN)</math>  <math>fmin(sNaN, sNaN) = fmin(sNaN, qNaN) = fmin(qNaN, sNaN) = quietize(sNaN)</math>  <math>fmin(qNaN, qNaN) = qNaN</math>  <math>fmin(+0, -0) = fmin(-0, +0) = -0</math>            Fmin with sNaN operand returns sNaN instead of quietize(sNaN)            [] Fmin(+0,-0) returns +0. Should be -0, to match EU Fmin instruction.         </td> </tr> <tr> <td>03h</td> <td>AOP3_FCMPWR</td> <td>new_dst = (src0 == old_dst) ? src1 : old_dst (default ternary opcode)</td> <td>The fcmpwr operation performs the comparison using IEEE specification rules, and performs the store as a raw move (so sNaN is not quietized).  <math>fcmpwr(NaN, x, y) = NaN</math>  <math>fcmpwr(x, NaN, y) = x</math>  <math>fcmpwr(x, x, NaN) = NaN</math> </td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	Programming Notes	01h	AOP2_FMAX	new_dst = fmax(old_dst, src0) (default binary opcode)	The fmax operation implements the IEEE specification, which differs slightly from the DX and OCL specifications when a source operand is a sNaN. $fmax(x, qNaN) = fmax(qNaN, x) = x$ $fmax(x, sNaN) = fmax(sNaN, x) = quietize(sNaN)$ $fmax(sNaN, sNaN) = fmax(sNaN, qNaN) = fmax(qNaN, sNaN) = quietize(sNaN)$ $fmax(qNaN, qNaN) = qNaN$ $fmax(-0, +0) = fmax(+0, -0) = +0$ Fmax with sNaN operand returns sNaN instead of quietize(sNaN)	02h	AOP2_FMIN	new_dst = fmin(old_dst, src0)	The fmin operation implements the IEEE specification, which differs slightly from the DX and OCL specifications when a source operand is a sNaN. $fmin(x, qNaN) = fmin(qNaN, x) = x$ $fmin(x, sNaN) = fmin(sNaN, x) = quietize(sNaN)$ $fmin(sNaN, sNaN) = fmin(sNaN, qNaN) = fmin(qNaN, sNaN) = quietize(sNaN)$ $fmin(qNaN, qNaN) = qNaN$ $fmin(+0, -0) = fmin(-0, +0) = -0$ Fmin with sNaN operand returns sNaN instead of quietize(sNaN) [] Fmin(+0,-0) returns +0. Should be -0, to match EU Fmin instruction.	03h	AOP3_FCMPWR	new_dst = (src0 == old_dst) ? src1 : old_dst (default ternary opcode)	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		
Value	Name	Description	Programming Notes																				
01h	AOP2_FMAX	new_dst = fmax(old_dst, src0) (default binary opcode)	The fmax operation implements the IEEE specification, which differs slightly from the DX and OCL specifications when a source operand is a sNaN. $fmax(x, qNaN) = fmax(qNaN, x) = x$ $fmax(x, sNaN) = fmax(sNaN, x) = quietize(sNaN)$ $fmax(sNaN, sNaN) = fmax(sNaN, qNaN) = fmax(qNaN, sNaN) = quietize(sNaN)$ $fmax(qNaN, qNaN) = qNaN$ $fmax(-0, +0) = fmax(+0, -0) = +0$ Fmax with sNaN operand returns sNaN instead of quietize(sNaN)																				
02h	AOP2_FMIN	new_dst = fmin(old_dst, src0)	The fmin operation implements the IEEE specification, which differs slightly from the DX and OCL specifications when a source operand is a sNaN. $fmin(x, qNaN) = fmin(qNaN, x) = x$ $fmin(x, sNaN) = fmin(sNaN, x) = quietize(sNaN)$ $fmin(sNaN, sNaN) = fmin(sNaN, qNaN) = fmin(qNaN, sNaN) = quietize(sNaN)$ $fmin(qNaN, qNaN) = qNaN$ $fmin(+0, -0) = fmin(-0, +0) = -0$ Fmin with sNaN operand returns sNaN instead of quietize(sNaN) [] Fmin(+0,-0) returns +0. Should be -0, to match EU Fmin instruction.																				
03h	AOP3_FCMPWR	new_dst = (src0 == old_dst) ? src1 : old_dst (default ternary opcode)	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																					
<b>Programming Notes</b>																							
Binary opcodes AOP2_FMAX, AOP2_FMIN, AOP2_FADD, AOP2_FSUB, AOP2_FADD_64b, and AOP2_FSUB_64b have one source data payload.																							
Ternary opcode AOP3_FCMPWR has two source data payloads.																							



## Atomic Float Ternary Operation Message Descriptor Control Field

MDC_FOP3 - Atomic Float Ternary Operation Message Descriptor Control Field															
Size (in bits):		3													
Default Value:		0x00000003													
DWord	Bit	Description													
0	2:0	<b>Atomic Float Operation Type</b> Specifies the atomic float ternary operation to be performed													
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>03h</td> <td>AOP_FCMPWR <b>[Default]</b></td> <td>new_dst = (src0 == old_dst) ? src1 : old_dst</td> <td>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</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	Programming Notes	03h	AOP_FCMPWR <b>[Default]</b>	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		
Value	Name	Description	Programming Notes												
03h	AOP_FCMPWR <b>[Default]</b>	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													
		<b>Programming Notes</b>													
		When Return Data Control is set, old_dst is returned.													

## Atomic Half Float Binary Operation Message Descriptor Control Field

<b>MDC_HFOP2 - Atomic Half Float Binary Operation Message Descriptor Control Field</b>				
Source:	BSpec			
Size (in bits):	3			
Default Value:	0x00000001			
DWord	Bit	Description		
0	2:0	<b>Atomic Float Operation Type</b> Specifies the atomic float binary operation to be performed		
		Value	Name	
		Description	Programming Notes	
		01h	<b>AOP_FMAX</b> <b>[Default]</b>	new_dst = fmax(old_dst, src0)  The fmax operation implements the IEEE specification, which differs slightly from the DX and OCL specifications 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(-0, +0) = fmax(+0, -0) = +0 Fmax with sNaN operand returns sNaN instead of quietize(sNaN)
		02h	<b>AOP_FMIN</b>	new_dst = fmin(old_dst, src0)  The fmin operation implements the IEEE specification, which differs slightly from the DX and OCL specifications 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(+0, -0) = fmin(-0, +0) = -0 Fmin with sNaN operand returns sNaN instead of quietize(sNaN) [] Fmin(+0,-0) returns +0. Should be -0, to match EU Fmin instruction.

## Atomic Integer Binary Operation Message Descriptor Control Field

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

<b>MDC_AOP - Atomic Integer Operation Message Descriptor Control Field</b>																																																								
Size (in bits):	4																																																							
Default Value:	0x00000000																																																							
DWord	Bit	Description																																																						
0	3:0	<p><b>Atomic Integer Operation Type</b> Specifies the atomic integer operation to be performed.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">05h</td> <td style="text-align: center;">AOP1_INC</td> <td>new_dst = old_dst + 1 (default unary opcode)</td> </tr> <tr> <td style="text-align: center;">06h</td> <td style="text-align: center;">AOP1_DEC</td> <td>new_dst = old_dst - 1</td> </tr> <tr> <td style="text-align: center;">0Fh</td> <td style="text-align: center;">AOP1_PREDEC</td> <td>new_dst = old_dst - 1</td> </tr> <tr> <td style="text-align: center;">01h</td> <td style="text-align: center;">AOP2_AND</td> <td>new_dst = old_dst AND src0 (default binary opcode)</td> </tr> <tr> <td style="text-align: center;">02h</td> <td style="text-align: center;">AOP2_OR</td> <td>new_dst = old_dst   src0</td> </tr> <tr> <td style="text-align: center;">03h</td> <td style="text-align: center;">AOP2_XOR</td> <td>new_dst = old_dst ^ src0</td> </tr> <tr> <td style="text-align: center;">04h</td> <td style="text-align: center;">AOP2_MOV</td> <td>new_dst = src0</td> </tr> <tr> <td style="text-align: center;">07h</td> <td style="text-align: center;">AOP2_ADD</td> <td>new_dst = old_dst + src0</td> </tr> <tr> <td style="text-align: center;">08h</td> <td style="text-align: center;">AOP2_SUB</td> <td>new_dst = old_dst - src0</td> </tr> <tr> <td style="text-align: center;">09h</td> <td style="text-align: center;">AOP2_REVSUB</td> <td>new_dst = src0 - old_dst</td> </tr> <tr> <td style="text-align: center;">0Ah</td> <td style="text-align: center;">AOP2_IMAX</td> <td>new_dst = imax(old_dst, src0)</td> </tr> <tr> <td style="text-align: center;">0Bh</td> <td style="text-align: center;">AOP2_IMIN</td> <td>new_dst = imin(old_dst, src0)</td> </tr> <tr> <td style="text-align: center;">0Ch</td> <td style="text-align: center;">AOP2_UMAX</td> <td>new_dst = umax(old_dst, src0)</td> </tr> <tr> <td style="text-align: center;">0Dh</td> <td style="text-align: center;">AOP2_UMIN</td> <td>new_dst = umin(old_dst, src0)</td> </tr> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">AOP3_CMPWR_2W</td> <td>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;">AOP3_CMPWR</td> <td>new_dst = (src0 == old_dst) ? src1 : old_dst (default ternary opcode)</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> </tbody> </table>	Value	Name	Description	05h	AOP1_INC	new_dst = old_dst + 1 (default unary opcode)	06h	AOP1_DEC	new_dst = old_dst - 1	0Fh	AOP1_PREDEC	new_dst = old_dst - 1	01h	AOP2_AND	new_dst = old_dst AND src0 (default binary opcode)	02h	AOP2_OR	new_dst = old_dst   src0	03h	AOP2_XOR	new_dst = old_dst ^ src0	04h	AOP2_MOV	new_dst = src0	07h	AOP2_ADD	new_dst = old_dst + src0	08h	AOP2_SUB	new_dst = old_dst - src0	09h	AOP2_REVSUB	new_dst = src0 - old_dst	0Ah	AOP2_IMAX	new_dst = imax(old_dst, src0)	0Bh	AOP2_IMIN	new_dst = imin(old_dst, src0)	0Ch	AOP2_UMAX	new_dst = umax(old_dst, src0)	0Dh	AOP2_UMIN	new_dst = umin(old_dst, src0)	00h	AOP3_CMPWR_2W	new_dst = (src0_2W == old_dst_2W) ? src1_2W : old_dst_2W	0Eh	AOP3_CMPWR	new_dst = (src0 == old_dst) ? src1 : old_dst (default ternary opcode)	Others	Reserved	Ignored
Value	Name	Description																																																						
05h	AOP1_INC	new_dst = old_dst + 1 (default unary opcode)																																																						
06h	AOP1_DEC	new_dst = old_dst - 1																																																						
0Fh	AOP1_PREDEC	new_dst = old_dst - 1																																																						
01h	AOP2_AND	new_dst = old_dst AND src0 (default binary opcode)																																																						
02h	AOP2_OR	new_dst = old_dst   src0																																																						
03h	AOP2_XOR	new_dst = old_dst ^ src0																																																						
04h	AOP2_MOV	new_dst = src0																																																						
07h	AOP2_ADD	new_dst = old_dst + src0																																																						
08h	AOP2_SUB	new_dst = old_dst - src0																																																						
09h	AOP2_REVSUB	new_dst = src0 - old_dst																																																						
0Ah	AOP2_IMAX	new_dst = imax(old_dst, src0)																																																						
0Bh	AOP2_IMIN	new_dst = imin(old_dst, src0)																																																						
0Ch	AOP2_UMAX	new_dst = umax(old_dst, src0)																																																						
0Dh	AOP2_UMIN	new_dst = umin(old_dst, src0)																																																						
00h	AOP3_CMPWR_2W	new_dst = (src0_2W == old_dst_2W) ? src1_2W : old_dst_2W																																																						
0Eh	AOP3_CMPWR	new_dst = (src0 == old_dst) ? src1 : old_dst (default ternary opcode)																																																						
Others	Reserved	Ignored																																																						
<b>Programming Notes</b>																																																								
<p>For Unary Opcodes, when Return Data Control is set, new_dst is returned by AOP_PREDEC and otherwise old_dst is returned.</p> <p>For Binary and Ternary Opcodes, when Return Data Control is set, old_dst is returned.</p> <p>Unary Opcodes AOP1_* do not have a source data payload. Binary Opcodes AOP2_* have one source data payload. Ternary Opcodes AOP3_* have two source data payloads.</p>																																																								

## Atomic Integer Ternary Operation Message Descriptor Control Field

<b>MDC_AOP3 - Atomic Integer Ternary Operation Message Descriptor Control Field</b>														
Size (in bits):		4												
Default Value:		0x0000000E												
DWord	Bit	Description												
0	3:0	<p><b>Atomic Integer Operation Type</b> Specifies the atomic integer ternary operation to be performed</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00h</td> <td style="text-align: center;">AOP_CMPWR_2W</td> <td style="text-align: center;"><math>\text{new\_dst} = (\text{src0\_2W} == \text{old\_dst\_2W}) ? \text{src1\_2W} : \text{old\_dst\_2W}</math></td> </tr> <tr> <td style="text-align: center;">0Eh</td> <td style="text-align: center;">AOP_CMPWR <b>[Default]</b></td> <td style="text-align: center;"><math>\text{new\_dst} = (\text{src0} == \text{old\_dst}) ? \text{src1} : \text{old\_dst}</math></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> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>When Return Data Control is set, old_dst is returned.</p>	Value	Name	Description	00h	AOP_CMPWR_2W	$\text{new\_dst} = (\text{src0\_2W} == \text{old\_dst\_2W}) ? \text{src1\_2W} : \text{old\_dst\_2W}$	0Eh	AOP_CMPWR <b>[Default]</b>	$\text{new\_dst} = (\text{src0} == \text{old\_dst}) ? \text{src1} : \text{old\_dst}$	Others	Reserved	Ignored
Value	Name	Description												
00h	AOP_CMPWR_2W	$\text{new\_dst} = (\text{src0\_2W} == \text{old\_dst\_2W}) ? \text{src1\_2W} : \text{old\_dst\_2W}$												
0Eh	AOP_CMPWR <b>[Default]</b>	$\text{new\_dst} = (\text{src0} == \text{old\_dst}) ? \text{src1} : \text{old\_dst}$												
Others	Reserved	Ignored												



## Atomic Integer Unary Operation Message Descriptor Control Field

MDC_AOP1 - Atomic Integer Unary Operation Message Descriptor Control Field																	
Size (in bits):		4															
Default Value:		0x00000005															
DWord	Bit	Description															
0	3:0	<p><b>Atomic Integer Operation Type</b> Specifies the atomic integer unary operation to be performed</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>05h</td> <td>AOP_INC <b>[Default]</b></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> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>When Return Data Control is set, new_dst is returned by AOP_PREDEC and otherwise old_dst is returned.</p>	Value	Name	Description	05h	AOP_INC <b>[Default]</b>	new_dst = old_dst + 1	06h	AOP_DEC	new_dst = old_dst - 1	0Fh	AOP_PREDEC	new_dst = old_dst - 1	Others	Reserved	Ignored
Value	Name	Description															
05h	AOP_INC <b>[Default]</b>	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														
Size (in bits):	2													
Default Value:	0x00000003													
DWord	Bit	Description												
0	1:0	<b>Power State</b>												
		<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 <b>[Default]</b></td> <td>D3</td> </tr> </tbody> </table>	Value	Name	Description	00b	D0	D0	01b,10b	Unsupported	Unsupported	11b	D3 <b>[Default]</b>	D3
		Value	Name	Description										
		00b	D0	D0										
01b,10b	Unsupported	Unsupported												
11b	D3 <b>[Default]</b>	D3												

## AVC CABAC

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



**AVC CABAC**

<b>AVC CABAC</b>		
	0	<b>MB Concealment Flag</b> Each pulse from this flag indicates one MB is concealed by hardware.

## AVC CAVLC

<b>AVC CAVLC</b>						
Source:	VideoCS					
Size (in bits):	16					
Default Value:	0x00000000					
DWord	Bit	Description				
0	15	<b>Total Zero out-of-bound Error</b> This flag indicates the Total zero SE count exceed the max number of coeffs allowed in an intra16x16 AC block.				
	14	<b>Coefficient level out-of-bound Error</b> This flag indicates the coded coefficient level SEs in the bit-stream is out-of-bound.				
	13	<b>RunBefore out-of-bound Error</b> This flag indicates the coded RunBefore SE value is larger than the remaining zero block count.				
	12	<b>Total coefficient Out-of-bound Error</b> This flag indicates the coded total coeff SE count exceed the max number of coeffs allowed in an intra16x16 AC block.				
	11	<b>Temporal Direction Motion Vector Out-of-Bound Error</b> This flag indicates motion vectors calculated from Temporal Direct Motion Vector is larger than the allowed range specified by the AVC spec.				
	10	<b>Reserved</b> <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	9	<b>Motion Vector Delta SE Out-of-Bound Error</b> This flag indicates inconsistent Motion Vector Delta SEs coded in the bit-stream.				
	8	<b>Reference Index SE Out-of-Bound Error</b> This flag indicates inconsistent Reference Index SEs coded in the bit-stream.				
	7	<b>RunBefore/TotalZero Error</b> This flag indicates one or more inconsistent RunBefore or TotalZero SEs coded in the bit-stream.				
	6	<b>Exponential Golomb Error</b> This flag indicates hardware detects more than 18 leadzero for skip and more than 19 for other SEs from the Exponential Golomb Logic				
	5	<b>Total Coeff SE Error</b> This flag indicates one or more inconsistent total coeff SEs coded in the bit-stream.				
4	<b>Macroblock Coded Block Pattern Error</b> This flag indicates inconsistent CBP SEs coded in the bit-stream.					
3	<b>Mbtype/submbtype Error</b> This flag indicates inconsistent MBtype/SubMBtype SEs coded in the bit-stream.					
2	<b>Chroma Intra prediction Mode Error</b> This flag indicates inconsistent Chroma Intra prediction mode SEs coded in the bit-stream.					

**AVC CAVLC**

	1	<b>Luma Intra prediction Mode Error</b> This flag indicates inconsistent luma Intra prediction mode SE coded in the bit-stream.
	0	<b>MB Concealment Flag</b> Each pulse from this flag indicates one MB is concealed by hardware.



## AVP\_PAK\_INSERT\_OBJECT\_INDIRECT\_PAYLOAD

AVP_PAK_INSERT_OBJECT_INDIRECT_PAYLOAD		
Source:	VideoCS	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	<b>Indirect Payload Data Size in bits</b> Format: U32 Number of bits to be inserted. Not including those skipped bytes in the beginning.
1..2	63:0	<b>Indirect Payload Base Address</b> Format: <b>SplitBaseAddress64ByteAligned</b> 48-bit address of the indirect payload data in memory buffer. <b>Programming Notes</b> Payload must begin in a byte position, but the payload can be ended in a bit position.
3	31:0	<b>Indirect Payload Base Address</b> Format: <b>MemoryAddressAttributes</b>

## AVP\_REF\_LIST\_ENTRY\_OLD

AVP_REF_LIST_ENTRY_OLD			
Source:	VideoCS		
Size (in bits):	32		
Default Value:	0x00000000		
DWord	Bit	Description	
0	31:15	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	14:0	<b>Reference Picture Frame ID[i]</b>	
		Format:	U15
		<p>This array corresponding to the array <b>Reference Picture Address (RefAddr[0-7])</b> defined in the AVP_PIPE_BUF_ADDR_STATE command.</p> <p>Frame ID is in decoding order (not display order).</p> <p>Frame ID is a unique number identifying a reference frame. It is 15-bit quantity and wraps around after <math>2^{15}-1</math></p>	

## Barrier Data Payload

<b>MDP_Barrier - Barrier Data Payload</b>		
Source:	EuSubFunctionGateway	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
2	31	<b>Reserved</b>
		Access: RO
		Format: MBZ
30:24	<b>Barrier ID</b>	Format: U7
		This field indicates which barrier state is updated. Range = [0,63]
23	<b>Predicate Mask Enable</b>	Format: Enable
		This bit indicates that the barrier is a predicated barrier and the SIMD channels passing the predicate should be summed. All threads sending this message to the same barrier should have an identical value for this field, and must specify a response length of 1 for the predicate sum response. Note that Global Barriers must not have the Predicate Mask Enable bit set.
		<b>Programming Notes</b>
		This control is intended only for GPGPU or Media threads. This control must not be set if the barrier is for a Hull Shader thread.
22:16	<b>Reserved</b>	Access: RO
		Format: MBZ
15	<b>Barrier Count Enable</b>	Format: Enable
		Allows the message to reprogram the terminating barrier count. If set, the stored value of the terminating barrier count is set to the value of Barrier Count field (below), and used for this barrier operation. If clear, the stored value of the terminating barrier count is not modified and the stored value is used for this barrier operation.
		<b>Programming Notes</b>
		This control is intended only for Hull Shader threads. This control must not be set if the barrier is allocated by a GPGPU or Media thread.

<b>MDP_Barrier - Barrier Data Payload</b>						
	14:8	<b>Barrier Count</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U7</td> </tr> </table> <p>If Barrier Count Enable is set, this field specifies the terminating barrier count. Otherwise this field is ignored. All threads that belong to a single barrier must deliver the same value for this field for a particular barrier iteration.</p>	Format:	U7		
	Format:	U7				
7:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
3	31:0	<b>Predicate Mask</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>This field has a bit set per SIMD channel that passes the predicate. For SIMD8 and SIMD16 the rest of the bits must be 0. This field is ignored for non-predicated barriers.</p>	Format:	U32		
Format:	U32					
4..7	127:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
		Access:	RO			
Format:	MBZ					



## BaseAddress4KByteAligned

BaseAddress4KByteAligned		
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..1	63:12	<b>Base Address</b>
		Format: VIRTUAL_ADDR[63:12]
	11:0	<b>Reserved</b>
		Access: RO
	Format: MBZ	



## Batch Buffer Stack Structure

<b>BATCH_BUFFER_STACK_STRUCTURE - Batch Buffer Stack Structure</b>				
Size (in bits):		192		
Default Value:		0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000		
<b>DWord</b>	<b>Bit</b>	<b>Description</b>		
0..5	191:0	<p><b>Batch Stack Entries</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>BATCH_STACK_ENTRY[3]</b></td> </tr> </table> <p>Stack containing details of the batch buffers currently in execution. The top of stack is determined by the Batch Buffer Stack Pointer.</p>	Format:	<b>BATCH_STACK_ENTRY[3]</b>
Format:	<b>BATCH_STACK_ENTRY[3]</b>			

## Batch Stack Entry

BATCH_STACK_ENTRY - Batch Stack Entry		
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:62	<b>Reserved</b>
		Access: RO
		Format: MBZ
	61:60	<b>Reserved</b>
		Access: RO
		Format: MBZ
	59	<b>POSH Start</b>
	Exists If: //POCS	
	58	<b>POSH Enable</b>
	Exists If: //RCS, POCS	
	This bit reflects the POSH Enable value programmed by the active MI_BATCH_BUFFER_START command.	
	57	<b>Address Space Indicator</b>
Format: U1		
This field reflects the effective address space indicator security level and may not be the same as the Address Space Indicator written using MI_BATCH_BUFFER_START.		
<b>Value</b>		<b>Name</b>
0h	GGTT <b>[Default]</b>	This Batch buffer is located in GGTT memory and is privileged.
1h	PPGTT	This Batch buffer is located in PPGTT memory and is non-privileged.
56	<b>Reserved</b>	
55:48	<b>Reserved</b>	
	Access: RO	
	Format: MBZ	
47:2	<b>Batch Buffer Head Pointer</b>	
	Format: GraphicsAddress[47:2]	
1:0	<b>Reserved</b>	
	Access: RO	
	Format: MBZ	

## 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	15:12	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	11	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	10:3	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	2	<b>Command Privilege Violation Error</b> 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	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
0	<b>Instruction Error</b> 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"> <li>Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported).</li> <li>Defeatured MI Instruction Opcodes:</li> </ul>		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	1		Instruction Error detected
	<b>Programming Notes</b>		
	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	<b>Reserved</b>
		Access: RO
		Format: MBZ
	23:16	<b>Binding Table Index</b>
		Format: U8 This field specifies the index of binding table entry that will be updated.
	15:0	<b>Surface State Pointer</b>
Format: SurfaceStateOffset[21:6]RENDER_SURFACE_STATE Surface State Pointer. This address points to a surface state block. This pointer is relative to the Surface State Base Address.		

## Bit Definition for Interrupt Control Registers - Media

Bit Definition for Interrupt Control Registers - Media			
Source:	VideoCS		
Size (in bits):	32		
Default Value:	0x00000000		
DWord	Bit	Description	
0	31:12	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	11	<b>Wait on Semaphore</b> 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. Ring Buffer Scheduling: Set when MI_SEMAPHORE_WAIT command is un-successful.	
	10	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	9	<b>Reserved</b>	
	8	<b>Context Switch Interrupt</b> Set when a context switch has just occurred. <b>Execlist Enable bit</b> needs to be set for this interrupt to occur.	
	7	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
6	<b>Timeout Counter Expired</b> Set when the VCS timeout counter has reached the timeout thresh-hold value.		
5	<b>Reserved</b>		
4	<b>MI_FLUSH_DW Notify Interrupt</b> 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	<b>Video Command Parser Error</b> 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.		
	<b>Page Table Error:</b> Indicates a page table error. <b>Instruction Parser Error:</b> The Blitter Instruction Parser encounters an error while parsing an		

## Bit Definition for Interrupt Control Registers - Media

		instruction.	
	2:1	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	0	<b>Video Command Parser User Interrupt</b> 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.	

## BLEND\_STATE\_ENTRY

BLEND_STATE_ENTRY											
Size (in bits):	64										
Default Value:	0x00000000, 0x00000000										
DWord	Bit	Description									
0..1	63	<b>Logic Op Enable</b> Format: Enable Enables the LogicOp function of the Pixel Processing pipeline.									
		<b>Programming Notes</b>									
		Enabling LogicOp and Color Buffer Blending at the same time is UNDEFINED									
62:59	62:59	<b>Logic Op Function</b> Format: <b>3D_Logic_Op_Function</b> 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.									
		<b>Reserved</b> Access: RO Format: MBZ									
36	36	<b>Pre-Blend Source Only Clamp Enable</b> Format: 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.									
		<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>Only Source(s) are clamped prior to blend function. Other inputs to blend must be clamped according to the behavior specified for "pre-blend color clamp disable" in the pre-blend color clamping table .</td> </tr> </tbody> </table>	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 be clamped according to the behavior specified for "pre-blend color clamp disable" in the pre-blend color clamping table .
		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 be clamped according to the behavior specified for "pre-blend color clamp disable" in the pre-blend color clamping table .									
<b>Programming Notes</b>											
This field is ignored (treated as DISABLED) for UINT and SINT RT surface formats. Blending is not supported for those RT surface formats. <b>When this bit is enabled Pre-Blend Color Clamp Enable must be disabled.</b>											

## BLEND\_STATE\_ENTRY

35:34	<b>Color Clamp Range</b>	<p>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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>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). Unsigned Floating Point components are clamped to positive zero.</td> </tr> <tr> <td>3</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table> <div style="text-align: center; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"><b>Programming Notes</b></div> <p>See table in Pre-Blending Color Clamp subsection for programming restrictions as a function of Blending and Pre-Blend Color Clamp.</p>	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). Unsigned Floating Point components are clamped to positive zero.	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). Unsigned Floating Point components are clamped to positive zero.															
3	Reserved	Reserved															
33	<b>Pre-Blend Color Clamp Enable</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">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" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 75%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>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> <div style="text-align: center; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"><b>Programming Notes</b></div> <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	<b>Post-Blend Color Clamp Enable</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">Enable</td> </tr> </table> <p>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> <div style="text-align: center; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"><b>Programming Notes</b></div> <p>This field is ignored (treated as DISABLED) for UINT and SINT RT surface formats. Blending is not supported for those RT surface formats. Post Blend Clamp Enable must be programmed identical to Pre Blend Clamp Enable. The device will automatically clamp source color channels to the respective RT surface range. <b>When this bit is enabled Pre-Blend Source Only Clamp</b></p>	Format:	Enable													
Format:	Enable																



<b>BLEND_STATE_ENTRY</b>					
	<b>Enable must be disabled.</b>				
31	<p><b>Color Buffer Blend Enable</b></p> <table border="1"> <tr> <td>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> <p style="text-align: center;"><b>Programming Notes</b></p> <p>Enabling LogicOp and ColorBufferBlending at the same time is UNDEFINED</p>	Format:	Enable		
Format:	Enable				
30:26	<p><b>Source Blend Factor</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Color_Buffer_Blend_Factor</b></td> </tr> </table> <p>Controls the "source factor" in the ColorBufferBlending function. Refer to Source Alpha Blend Factor for encodings.</p>	Format:	<b>3D_Color_Buffer_Blend_Factor</b>		
Format:	<b>3D_Color_Buffer_Blend_Factor</b>				
25:21	<p><b>Destination Blend Factor</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Color_Buffer_Blend_Factor</b></td> </tr> </table> <p>Controls the "destination factor" in the ColorBufferBlending function. Refer to Source Alpha Blend Factor for encodings.</p>	Format:	<b>3D_Color_Buffer_Blend_Factor</b>		
Format:	<b>3D_Color_Buffer_Blend_Factor</b>				
20:18	<p><b>Color Blend Function</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Color_Buffer_Blend_Function</b></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:	<b>3D_Color_Buffer_Blend_Function</b>		
Format:	<b>3D_Color_Buffer_Blend_Function</b>				
17:13	<p><b>Source Alpha Blend Factor</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Color_Buffer_Blend_Factor</b></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:	<b>3D_Color_Buffer_Blend_Factor</b>		
Format:	<b>3D_Color_Buffer_Blend_Factor</b>				
12:8	<p><b>Destination Alpha Blend Factor</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Color_Buffer_Blend_Factor</b></td> </tr> </table> <p>Controls the "destination factor" in alpha Color Buffer Blending stage. Refer to Source Alpha Blend Factor for encodings.</p>	Format:	<b>3D_Color_Buffer_Blend_Factor</b>		
Format:	<b>3D_Color_Buffer_Blend_Factor</b>				
7:5	<p><b>Alpha Blend Function</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Color_Buffer_Blend_Function</b></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:	<b>3D_Color_Buffer_Blend_Function</b>		
Format:	<b>3D_Color_Buffer_Blend_Function</b>				
4	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				
3	<p><b>Write Disable Alpha</b></p> <table border="1"> <tr> <td>Format:</td> <td>Disable</td> </tr> </table> <p>This field controls the writing of the alpha component into the Render Target.</p>	Format:	Disable		
Format:	Disable				

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

## BLEND\_STATE

<b>BLEND_STATE</b>				
Size (in bits):	544			
Default Value:	0x00000000, 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><b>Alpha To Coverage Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">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><b>Independent Alpha Blend Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">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><b>Alpha To One Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">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><b>Alpha To Coverage Dither Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">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

<b>BLEND_STATE</b>																	
1..16	<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">27</td> <td style="width: 15%;"><b>Alpha Test Enable</b></td> <td style="width: 40%;">Format:</td> <td style="width: 40%; 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;"><b>Programming Notes</b></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	<b>Alpha Test Enable</b>	Format:	Enable	<p>Enables the AlphaTest function of the Pixel Processing pipeline. The field is applied to all the RTs in MRT case.</p>				<b>Programming Notes</b>				<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	<b>Alpha Test Enable</b>	Format:	Enable													
	<p>Enables the AlphaTest function of the Pixel Processing pipeline. The field is applied to all the RTs in MRT case.</p>																
	<b>Programming Notes</b>																
	<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>																
	26:24	<b>Alpha Test Function</b>	Format:	<b>3D_Compare_Function</b>													
	<p>This field specifies the comparison function used in the AlphaTest function. The field is applied to all the RTs in MRT case.</p>																
23	<b>Color Dither Enable</b>	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>																	
<b>Programming Notes</b>																	
<p>For YUV render target formats, this field must be programmed to 0.</p>																	
22:21	<b>X Dither Offset</b>	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>																	
20:19	<b>Y Dither Offset</b>	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>																	
18:0	<b>Reserved</b>	Access:	RO														
		Format:	MBZ														
511:0	<b>Entry</b>	Format:	<b>BLEND_STATE_ENTRY[8]</b>														

## Blitter Interrupt Vector

BLITTER_INTR_VEC - Blitter Interrupt Vector			
Size (in bits):	16		
Default Value:	0x00000000		
DWord	Bit	Description	
0	15	<b>Catastrophic Error</b> This interrupt signals that a unrecoverable error (for e.g encountered fault when accessing a page mapped in Global GTT) during the engine processing. When Memory interface signals this error, the Command Streamer will stop parsing any more instructions. Scheduler is expected to reset the engine to evict the context	
	14:12	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	11	<b>BCS Wait On Semaphore</b>	
	10	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	9	<b>CS TR Invalid Tile Detection</b>	
	8	<b>BCS Context Switch Interrupt</b>	
	7	<b>Legacy Context Per Process Page Fault Interrupt</b> Fault interrupt is generated by GA fabric, not by the CS This interrupt is for handling Legacy context PPGTT Page Fault.	
	6	<b>BCS Watchdog Counter Expired</b>	
	5	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
4	<b>BCS MI Flush DW Notify</b>		
3	<b>BCS Error Interrupt</b>		
2:1	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
0	<b>BCS MI User Interrupt</b>		

## Block Dimensions Message Header Control

MHC_BDIM - Block Dimensions Message Header Control				
Size (in bits):		32		
Default Value:		0x00000000		
DWord	Bit	Description		
0	31:22	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	21:20	<b>Block Height</b>		
		Height in rows of block being accessed. Range = [0,3] representing 1 to 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	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	1:0	<b>Block Width</b>		
		Width in Dwords of block being accessed. Range = [0,3] representing 1 to 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

<b>MH_BTS_GO - Block Message Header</b>		
Source:	EuSubFunctionDataPort0	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
2	31:0	<b>Global Offset</b>
		Format: U32
		Specifies the global element index into the buffer, in units of Hwords, Owords, Dwords, or Bytes (depending on the message).
		<b>Programming Notes</b>
		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	<b>Reserved</b>
		Access: RO
		Format: MBZ

## BR00 - BLT Opcode and Control

BR00 - BLT Opcode and Control										
Source:	BlitterCS									
Size (in bits):	32									
Default Value:	0x00000000									
DWord	Bit	Description								
0	31	<b>BLT Engine Busy</b> 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"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Idle <b>[Default]</b></td> </tr> <tr> <td>1</td> <td>Busy</td> </tr> </tbody> </table>	Value	Name	0	Idle <b>[Default]</b>	1	Busy		
		Value	Name							
	0	Idle <b>[Default]</b>								
	1	Busy								
	<table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> </table>	Default Value:	0							
Default Value:	0									
30	<b>Setup Instruction Instruction</b> The current instruction performs clipping (1).									
29	<b>Setup Monochrome Pattern</b> 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"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Color <b>[Default]</b></td> </tr> <tr> <td>1</td> <td>Monochrome</td> </tr> </tbody> </table>	Value	Name	0	Color <b>[Default]</b>	1	Monochrome		
		Value	Name							
		0	Color <b>[Default]</b>							
1	Monochrome									
<table border="1"> <tr> <td>Default Value:</td> <td>0000000b</td> </tr> </table>	Default Value:	0000000b								
Default Value:	0000000b									
28:22	<b>Instruction Target (Opcode)</b> 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.									
21:20	<b>32bpp Byte Mask</b> This field is only used for 32bpp.	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td><b>[Default]</b></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	<b>[Default]</b>	1xb	Write Alpha Channel	x1b	Write RGB Channel
		Value	Name							
		00b	<b>[Default]</b>							
		1xb	Write Alpha Channel							
x1b	Write RGB Channel									
<table border="1"> <tr> <td>Default Value:</td> <td>000b</td> </tr> </table>	Default Value:	000b								
Default Value:	000b									
19:17	<b>Monochrome Source Start</b> 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.									



<b>BR00 - BLT Opcode and Control</b>		
16	<b>Bit/Byte Packed</b>	
	Byte packed is for the NT driver.	
	<b>Value</b>	<b>Name</b>
	0b	Bit <b>[Default]</b>
	1b	Byte
15	<b>Src Tiling Enable</b>	
	<b>Value</b>	<b>Name</b>
	0b	Tiling Disabled (Linear) <b>[Default]</b>
	1b	Tiling enabled: Tile-X or Tile-Y
14:12	<b>Horizontal Pattern Seed</b>	
	Default Value:	0b
	This field indicates the pattern pixel position which corresponds to X = 0.	
11	<b>Dest Tiling Enable</b>	
	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.	
	<b>Value</b>	<b>Name</b>
	0b	Tiling Disabled (Linear blit) <b>[Default]</b>
	1b	Tiling enabled: Tile-X or Tile-Y
10:8	<b>Transparency Range Mode</b>	
	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.	
	<b>Value</b>	<b>Name</b>
	xx0b	<b>[Default]</b>
	No color transparency mode enabled. This causes normal operation with regard to writing data to the destination.	
001b		[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.
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

## BR00 - BLT Opcode and Control

		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	<b>Pattern Vertical Seed</b>	
	Default Value:	000b
	This field specifies the pattern scan line which corresponds to Y=0.	
4	<b>Destination Read Modify Write</b>	
	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	<b>Color Source</b>	
	Default Value:	0b
	This bit is decoded from the last instructions opcode field to identify whether a color (1) source is used.	
2	<b>Monochrome Source</b>	
	Default Value:	0b
	This bit is decoded from the last instructions opcode field to identify whether a monochrome (1) source is used.	
1	<b>Color Pattern</b>	
	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	<b>Monochrome Pattern</b>
		Default Value: 0b
		This bit is decoded from the last instructions opcode field to identify whether a monochrome (1) pattern is used.

## 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><b>Solid Pattern Select</b></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><b>Clipping Enabled</b></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><b>Monochrome Source Transparency Mode</b></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	<p><b>Monochrome Pattern Transparency Mode</b></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 field.</p> <table border="1" data-bbox="334 491 1464 877"> <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. 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>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	<p><b>32bpp Byte Mask</b></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 field.</p> <table border="1" data-bbox="334 1098 1464 1281"> <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										
25:24	<p><b>Color Depth</b></p> <table border="1" data-bbox="334 1325 1464 1556"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 Bit Color Depth [Default]</td> </tr> <tr> <td>01b</td> <td>16 Bit Color Depth</td> </tr> <tr> <td>10b</td> <td>Alternate 16 Bit Color Depth</td> </tr> <tr> <td>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	<p><b>Raster Operation Select</b></p> <p>These 8 bits are used to select which one of 256 possible raster operations is to be performed by the BLT Engine.</p>										
15:0	<p><b>Destination Pitch (Offset)</b></p> <p>For non-XY Blits, the signed 16bit field allows for specifying up to + 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 up to + 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 up to + 32KWords. For X, Y blits with nontiled surfaces (linear surfaces), this 16bit field can be programmed to byte</p>										

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

		<p>specification of up to + 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	<p><b>Setup Expansion Background Color Bits</b></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 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.</p>

## 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><b>Setup Expansion Foreground Color Bits</b></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><b>Setup Blit Color Pattern Address</b></p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[31:6]</td> </tr> </table> <p>Lower 32bits of the 48bit addressing.</p> <p>These 26 bits specify the starting address of the (8X8) pixel color <b>pattern from the SETUP_BLT instruction</b>. This register works identically to the Pattern Address register (BR15), but this version is <b>only used with the SCANLINE_BLT instruction execution</b> (the actual programming for this, is done in XY_SETUP_BLT command). The pattern data must be located in linear memory.</p> <p>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.</p> <p>The Pattern Base Address programmed, must always be Cache Line (64byte) aligned.</p>	Format:	GraphicsAddress[31:6]	
	Format:	GraphicsAddress[31:6]			
5:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
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><b>Destination Address Bits</b></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	<b>Reserved</b>
	15:0	<p><b>Source Pitch (Offset)</b></p> <p>For non-XY Blits with color source operand (SRC_COPY_BLT), the signed 16bit field allows for specifying up to + 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 up to + 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 up to + 32KDWords. For X, Y blits with nontiled color source surfaces (linear surfaces), this 16bit field can be programmed to byte specification of up to + 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><b>Source Address Bits</b></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 aligned 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><b>Solid Pattern Select</b></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><b>Clipping Enabled</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> </table>	Default Value:	0								
Default Value:	0										
29		<p><b>Monochrome Source Transparency Mode</b></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.									
28	<p><b>Monochrome Pattern Transparency Mode</b></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</p>										

## BR13 - BLT Raster OP, Control, and Destination Pitch

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.

Value	Name	Description
0	<b>[Default]</b>	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><b>32bpp Byte Mask</b> This field is only used for 32bpp.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td style="text-align: center;"><b>[Default]</b></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	<b>[Default]</b>	1xb	Write Alpha Channel	x1b	Write RGB Channel
Value	Name								
00b	<b>[Default]</b>								
1xb	Write Alpha Channel								
x1b	Write RGB Channel								

25:24	<p><b>Color Depth</b></p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00b</td> <td>8 Bit Color Depth <b>[Default]</b></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>24 Bit Color Depth</td> </tr> <tr> <td style="text-align: center;">11b</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	00b	8 Bit Color Depth <b>[Default]</b>	01b	16 Bit Color Depth	10b	24 Bit Color Depth	11b	Reserved
Value	Name										
00b	8 Bit Color Depth <b>[Default]</b>										
01b	16 Bit Color Depth										
10b	24 Bit Color Depth										
11b	Reserved										

23:16	<p><b>Raster Operation Select</b></p> <table border="1"> <tr> <td>Default Value:</td> <td style="text-align: center;">00000000b</td> </tr> </table> <p>These 8 bits are used to select which one of 256 possible raster operations is to be performed by the BLT Engine.</p>	Default Value:	00000000b
Default Value:	00000000b		

15:0	<p><b>Destination Pitch(Offset)</b></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	
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.		
DWord	Bit	Description
0	31:29	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:16	<b>Destination Height</b> 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	<b>Reserved</b>	
	Access: RO	
	Format: MBZ	
12:0	<b>Destination Byte Width</b> 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><b>Color Pattern Address</b></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><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
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><b>Pattern Expansion Background Color Bits</b></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	<p><b>Pattern Expansion Background Color Bits</b></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 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>

## 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><b>Source Expansion Background Color Bits</b></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	<p><b>Pattern/Source Expansion Foreground Color Bits</b></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 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.</p>

## BR27 - Destination Higher Order Address

BR27 - Destination Higher Order Address			
Source:	BlitterCS		
Size (in bits):	32		
Default Value:	0x00000000		
<p>Upper 32 bits of the starting pixel address for the destination data. This structure is also the working location for the upper bits of the destination address, and changes as the BLT Engine performs the accesses. See BR09 for the lower 32 bits. 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.</p> <p>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.</p> <p>Note that for non-XY blits (COLOR_BLT, SRC_COPY_BLT), the destination address points to the first byte to be written. This structure is always the last location written for a BLT drawing instruction. Writing to 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> <p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The uppermost reserved bits are ignored and MBZ.</p>			
DWord	Bit	Description	
0	31:25	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	24:16	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	15:0	<b>Destination Address Upper DWORD</b>	
		Format: GraphicsAddress[47:32]	

## BR28 - Source Higher Order Address

BR28 - Source Higher Order Address						
Source:	BlitterCS					
Size (in bits):	32					
Default Value:	0x00000000					
DWord	Bit	Description				
<p>0</p> <p>Upper 32 bits of the Source address, specifying 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> <p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The uppermost reserved bits are ignored and MBZ.</p>	31:25	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
24:16	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
15:0	<p><b>Source Address Upper DWORD</b></p> <table border="1"> <tr> <td>Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table>	Format:	GraphicsAddress[47:32]			
Format:	GraphicsAddress[47:32]					

## BR29 - Color Pattern Higher Order Address

BR29 - Color Pattern Higher Order Address		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
<p>0</p> <p>Upper 32 bits of the Color Pattern address, specifying the starting location of the (8X8) pixel pattern. GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The uppermost reserved bits are ignored and MBZ.</p>	31:25	<b>Reserved</b>
		Access:
	Format:	MBZ
	24:16	<b>Reserved</b>
		Access:
	Format:	MBZ
15:0	<b>Color Pattern Address Upper DWORD</b>	
	Format:	GraphicsAddress[47:32]



## BR30 - Setup Blit Color Pattern Higher Order Address

BR30 - Setup Blit Color Pattern Higher Order Address		
Source:	BlitterCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:16	<b>Reserved</b>
Upper 32 bits of the Color Pattern address, specifying the starting location of the (8X8) pixel pattern. GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The uppermost reserved bits are ignored and MBZ.		Access:
		Format:
	15:0	<b>Setup Blit Color Pattern Upper DWORD</b>
		Format:



## Byte Masked Media Block Message Header

MH_MBBM - Byte Masked Media Block Message Header		
Source:	EuSubFunctionDataPort1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	<b>X Offset</b>
		Format: S31
		X offset (in bytes) of the upper left corner of the block into the surface.
		<b>Programming Notes</b> Must be DWord aligned (Bits 1:0 MBZ) for the write form of the message.
1	31:0	<b>Y Offset</b>
		Format: S31 Y offset (in rows) of the upper left corner of the block into the surface.
2	31:0	<b>Media Block Message Control</b>
		Format: <b>MHC_MBBM_CONTROL</b> Specifies the Byte Masked message subtype and its additional input parameters.
3	31:0	<b>Byte Mask</b>
		Format: U32
		Specifies the Byte Mask for writes when Message Mode field is BYTE_MASK.
		<b>Programming Notes</b> The Byte mask applies horizontally to each row of output: bit 0 for byte 0, through bit 31 for byte 31.
4	31:0	<b>FFTID</b>
		Format: <b>MHC_FFTID</b> Fixed Function Thread ID
5..7	95:0	<b>Reserved</b>
		Access: RO Format: MBZ

## Byte Masked Media Block Message Header Control

MHC_MBBM_CONTROL - Byte Masked Media Block Message Header Control											
Size (in bits):		32									
Default Value:		0x00000000									
DWord	Bit	Description									
0	31:30	<b>Message Mode</b> Specifies the Media Block Write Message subtype is Byte Masked. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>02h</td> <td>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>Others</td> <td>Reserved</td> <td>Reserved.</td> </tr> </tbody> </table>	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.
		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	<b>Reserved</b> Access: RO Format: MBZ									
		28:24 <b>Sub-Register Offset</b> Format: U5 This field is ignored (reserved) for Media Block Write message.									
	23:22	<b>Reserved</b> Access: RO Format: MBZ									
		21:16 <b>Block Height</b> Format: U6 Height in rows of block being accessed. Range = [0,63] representing 1 to 64 rows <table border="1"> <thead> <tr> <th>Restriction</th> </tr> </thead> <tbody> <tr> <td>If Block Width (bytes), then Maximum Block Height (rows) is constrained by (# Dwords width) * (# rows) &lt;= 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.										
15:10	<b>Reserved</b> Access: RO Format: MBZ										
	9:8 <b>Register Pitch Control</b> Format: U2 This field is ignored (reserved) for a Media Block Write message.										

## MHC\_MBBM\_CONTROL - Byte Masked Media Block Message Header Control

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

## CC\_VIEWPORT

<b>CC_VIEWPORT</b>					
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 and Maximum Depth legal value ranges are dependent on the depth buffer format.</p>					
DWord	Bit	Description			
0	31:0	<p><b>Minimum Depth</b></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;"><b>Programming Notes</b></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).            For All depth formats: Minimum depth value must not be less than 0.0, also it may not be -0.0 (negative zero)</p>	Format:	IEEE_FLOAT	<b>Programming Notes</b>
Format:	IEEE_FLOAT				
<b>Programming Notes</b>					
1	31:0	<p><b>Maximum Depth</b></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;"><b>Programming Notes</b></td> </tr> </table> <p>The Maximum depth value cannot be smaller than Minimum depth value.            The Maximum depth value cannot be NAN (Not-A-Number).            For all depth formats: The Maximum depth value must be between +0.0to +1.0.</p>	Format:	IEEE_FLOAT	<b>Programming Notes</b>
Format:	IEEE_FLOAT				
<b>Programming Notes</b>					

## Channel Mask Message Descriptor Control Field

MDC_CMASK - Channel Mask Message Descriptor Control Field																																																					
Size (in bits):	4																																																				
Default Value:	0x00000000																																																				
DWord	Bit	Description																																																			
0	3:0	<p><b>Mask</b> 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 <b>[Default]</b></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>	Value	Name	Description	00h	RGBA <b>[Default]</b>	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
Value	Name	Description																																																			
00h	RGBA <b>[Default]</b>	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

<b>MDC_CMODE - Channel Mode Message Descriptor Control Field</b>		
Size (in bits):		1
Default Value:		0x00000000
DWord	Bit	Description
0	0	<b>Channel Mode</b> Format: Boolean 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.
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.

## Clear Color

<b>CLEAR_COLOR - Clear Color</b>		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	<b>Raw Clear Color : Red</b>
<b>Programming Notes:</b> Software shall write the Raw Clear Color channels such that the channel order matches the SURFACE_STATE.Shader Channel Select programming.  Software shall write the converted Depth Clear to this dword		Format: IEEE_FLOAT
		Format: U32
		Format: S31
		Format: <b>U24_X8 []</b>
1	31:0	<b>Raw Clear Color: Blue</b>
<b>Programming Notes:</b> Software shall write the Raw Clear Color channels such that the channel order matches the SURFACE_STATE.Shader Channel Select programming.		Format: IEEE_FLOAT
		Format: U32
		Format: S31
2	31:0	<b>Raw Clear Color : Green</b>
<b>Programming Notes:</b> Software shall write the Raw Clear Color channels such that the channel order matches the SURFACE_STATE.Shader Channel Select programming.		Format: IEEE_FLOAT
		Format: U32
		Format: S31
3	31:0	<b>Raw Clear Color : Alpha</b>
<b>Programming Notes:</b> Software shall write the Raw Clear Color channels such that the channel order matches the SURFACE_STATE.Shader Channel Select programming.		Format: IEEE_FLOAT
		Format: U32
		Format: S31
4	31:0	<b>Converted Clear Color and Clear Depth</b> This DWORD stores the format converted clear color. If bits per pixel are 32, entire pixel's clear value is stored in this DWORD. If bits per pixel are 64, lower DOWRD is stored in this field. If bits per pixel are 128, this field is not used to store clear value. This field is packed according to the RT format

<b>CLEAR_COLOR - Clear Color</b>					
5	31:0 <b>Converted Clear Color</b> This DWORD stores the format converted clear color. If bits per pixel are 64, upper DOWRD is stored in this field If bits per pixel are 32 or 128, this field is not used to store clear value. The field is packed according to the RT format				
6	31:1 <b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO			
Format:	MBZ				
	0 <b>Color Discard Enable</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; background-color: #e6f2ff;">Description</th> </tr> <tr> <td>When this bit is set for a Render Target Surface, SW indicates HW that this surface's cachelines from on-chip caches do not need to be written back to memory after a complete Render Pass (aka Tile Pass). This bit applies to both the main and the associated AUX surfaces.</td> </tr> <tr> <th style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td>This bit must be programmed before binding a surface to a Render Pass (Tile Pass). This bit must not be changed during the Tile Pass.</td> </tr> </table>	Description	When this bit is set for a Render Target Surface, SW indicates HW that this surface's cachelines from on-chip caches do not need to be written back to memory after a complete Render Pass (aka Tile Pass). This bit applies to both the main and the associated AUX surfaces.	Programming Notes	This bit must be programmed before binding a surface to a Render Pass (Tile Pass). This bit must not be changed during the Tile Pass.
Description					
When this bit is set for a Render Target Surface, SW indicates HW that this surface's cachelines from on-chip caches do not need to be written back to memory after a complete Render Pass (aka Tile Pass). This bit applies to both the main and the associated AUX surfaces.					
Programming Notes					
This bit must be programmed before binding a surface to a Render Pass (Tile Pass). This bit must not be changed during the Tile Pass.					
7	31:1 <b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO			
Format:	MBZ				
	0 <b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				



## Clock Gating Disable Format

Clock Gating Disable Format											
Size (in bits):		1									
Default Value:		0x00000000									
DWord	Bit	Description									
0	0	<b>Clock_Gate_Disable</b>									
		<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									

## COLOR\_CALC\_STATE

COLOR_CALC_STATE											
Size (in bits):	192										
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000										
This definition applies to 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	<b>Reserved</b>									
		Access: RO Format: MBZ									
	15	<b>Round Disable Function Disable</b>									
		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	<b>Reserved</b>										
	Access: RO Format: MBZ										
	0 <b>Alpha Test Format</b> 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									
<p style="text-align: center;"><b>Programming Notes</b></p> <p>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.</p>											
1	31:0	<b>Alpha Reference Value As FLOAT32</b>									
		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.										
	<p style="text-align: center;"><b>Programming Notes</b></p> <p>This field should not be programmed to NaN.</p>										

<b>COLOR_CALC_STATE</b>						
	7:0	<p><b>Alpha Reference Value As UNORM8</b></p> <table border="1"> <tr> <td>Exists If:</td> <td>[Alpha Test Format] == 'ALPHATEST_UNORM8'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> <p>This field specifies the alpha reference value to compare against in the Alpha Test function.</p>	Exists If:	[Alpha Test Format] == 'ALPHATEST_UNORM8'	Format:	UNORM8
Exists If:	[Alpha Test Format] == 'ALPHATEST_UNORM8'					
Format:	UNORM8					
2	31:0	<p><b>Blend Constant Color Red</b></p> <table border="1"> <tr> <td>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		
Format:	IEEE_FLOAT					
3	31:0	<p><b>Blend Constant Color Green</b></p> <table border="1"> <tr> <td>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		
Format:	IEEE_FLOAT					
4	31:0	<p><b>Blend Constant Color Blue</b></p> <table border="1"> <tr> <td>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		
Format:	IEEE_FLOAT					
5	31:0	<p><b>Blend Constant Color Alpha</b></p> <table border="1"> <tr> <td>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		
Format:	IEEE_FLOAT					

## COLOR\_PROCESSING\_STATE - ACE State

COLOR_PROCESSING_STATE - ACE State							
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	<b>Reserved</b>					
		Access: RO					
		Format: MBZ					
	6:2	<b>Skin Threshold</b> 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
Value	Name						
1-31							
26	[Default]						
1	<b>Full Image Histogram</b> Default Value: 0 Format: Enable Used to ignore the area of interest for full image histogram.						
0	<b>ACE Enable</b> Format: Enable						
1	31:24	<b>Y3</b> Default Value: 76 Format: U8 The value of the y_pixel for point 3 in PWL.					
	23:16	<b>Y2</b> Default Value: 56 Format: U8 The value of the y_pixel for point 2 in PWL.					

<b>COLOR_PROCESSING_STATE - ACE State</b>						
	15:8	<b>Y1</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">36</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>The value of the y_pixel for point 1 in PWL.</p>	Default Value:	36	Format:	U8
	Default Value:	36				
Format:	U8					
7:0	<b>Ymin</b> <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	<b>Y7</b> <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	<b>Y6</b> <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	<b>Y5</b> <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	<b>Y4</b> <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	<b>Ymax</b> <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	<b>Y10</b> <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					

COLOR_PROCESSING_STATE - ACE State						
	15:8	<b>Y9</b> <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	<b>Y8</b> <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	<b>B4</b> <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	<b>B3</b> <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					
15:8	<b>B2</b> <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	<b>B1</b> <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	<b>B8</b> <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	<b>B7</b> <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					

COLOR_PROCESSING_STATE - ACE State						
	15:8	<b>B6</b> <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	<b>B5</b> <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	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
15:8	<b>B10</b> <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					
7:0	<b>B9</b> <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 bias for point 9 in PWL.</p>	Default Value:	196	Format:	U8	
Default Value:	196					
Format:	U8					
7	31:27	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	26:16	<b>S1</b> <table border="1"> <tr> <td>Format:</td> <td>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					
15:11	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
10:0	<b>S0</b> <table border="1"> <tr> <td>Format:</td> <td>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					

<b>COLOR_PROCESSING_STATE - ACE State</b>						
8	31:27	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	26:16	<b>S3</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">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					
15:11	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
10:0	<b>S2</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">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	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	26:16	<b>S5</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">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					
15:11	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
10:0	<b>S4</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">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	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
26:16	<b>S7</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;">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	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					



<b>COLOR_PROCESSING_STATE - ACE State</b>						
	10:0	<p><b>S6</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>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><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	26:16	<p><b>S9</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>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><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
	10:0	<p><b>S8</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>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><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
	10:0	<p><b>S10</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>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 - PROCAMP State

COLOR_PROCESSING_STATE - PROCAMP State		
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	<b>Reserved</b>
		Access: RO
		Format: MBZ
	27:17	<b>Contrast</b>
		Default Value: 1
Format: U4.7 Contrast magnitude.		
16:13	<b>Reserved</b>	
	Access: RO	
	Format: MBZ	
12:1	<b>Brightness</b>	
	Default Value: 0	
	Format: S7.4 Brightness magnitude.	
0	<b>PROCAMP Enable</b>	
	Default Value: 1	
	Format: Enable	
1	31:16	<b>Cos_c_s</b>
		Default Value: 256
		Format: S7.8 UV multiplication cosine factor.
	15:0	<b>Sin_c_s</b>
		Default Value: 0
Format: S7.8 UV multiplication sine factor.		

## COLOR\_PROCESSING\_STATE - STD/STE State

COLOR_PROCESSING_STATE - STD/STE State			
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	<b>V_Mid</b>	
		Default Value:	154
		Format:	U8
			Rectangle middle-point V coordinate
	23:16	<b>U_Mid</b>	
		Default Value:	110
		Format:	U8
			Rectangle middle-point U coordinate
	15:10	<b>Hue Max</b>	
		Default Value:	14
Format:		U6	
		Rectangle half width	
9:4	<b>Sat Max</b>		
	Default Value:	31	
	Format:	U6	
		Rectangle half length.	
3	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
2	<b>Output Control</b>		
	<b>Value</b>	<b>Name</b>	
	0	Output Pixels <b>[Default]</b>	
	1	Output STD Decisions	

COLOR_PROCESSING_STATE - STD/STE State		
	1	<b>STE Enable</b> Format: Enable
	0	<b>STD Enable</b> Format: Enable
1	31	<b>Reserved</b> Access: RO Format: MBZ
	30:28	<b>Diamond Margin</b> Default Value: 4 Format: U3
	27:21	<b>Diamond du</b> Default Value: 0 Format: S6 Rhombus center shift in the sat-direction, relative to the rectangle center.
	20:18	<b>HS Margin</b> Default Value: 3 Format: U3
	17:10	<b>Cos(<math>\hat{I}\pm</math>)</b> Format: S0.7 The default is 79/128
	9:8	<b>Reserved</b> Access: RO Format: MBZ
	7:0	<b>Sin(<math>\hat{I}\pm</math>)</b> Format: S0.7 The default is 101/128
	2	31:21
20:13		<b>Diamond Alpha</b> Format: U2.6 1 / tan() The default is 100/64

COLOR_PROCESSING_STATE - STD/STE State						
	12:7	<b>Diamond Th</b> <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	<b>Diamond dv</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table>	Default Value:	0	Format:	S6	
Default Value:	0					
Format:	S6					
3	31:24	<b>Y_point_3</b> <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	<b>Y_point_2</b> <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	<b>Y_point_1</b> <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	<b>VY_STD_Enable</b> <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	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
4	31:18	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
17:13	<b>Y_Slope_2</b> <table border="1"> <tr> <td>Format:</td> <td>U2.3</td> </tr> </table> <p>Slope between points Y3 and Y4. The default is 31/8.</p>	Format:	U2.3			
Format:	U2.3					

COLOR_PROCESSING_STATE - STD/STE State										
	12:8	<b>Y_Slope_1</b> <table border="1"> <tr> <td>Format:</td> <td>U2.3</td> </tr> </table> <p>Slope between points Y1 and Y2. The default is 31/8.</p>	Format:	U2.3						
	Format:	U2.3								
7:0	<b>Y_point_4</b> <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									
5	31:16	<b>INV_skin_types_margin</b> <table border="1"> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p><math>1/(2 * \text{Skin\_types\_margin})</math></p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>20</td> <td>[Default]</td> <td>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	<b>Inverse Margin VYL</b> <table border="1"> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p><math>1 / \text{Margin\_VYL}</math> The default is 3300/65536</p>	Format:	U0.16							
Format:	U0.16									
6	31:24	<b>P1L</b> <table border="1"> <tr> <td>Default Value:</td> <td>216</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Y Point 1 of the lower part of the detection PWLF.</p>	Default Value:	216	Format:	U8				
	Default Value:	216								
	Format:	U8								
23:16	<b>P0L</b> <table border="1"> <tr> <td>Default Value:</td> <td>46</td> </tr> <tr> <td>Format:</td> <td>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	<b>Inverse Margin VYU</b> <table border="1"> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> <p><math>1 / \text{Margin\_VYU}</math> The default is 1600/65536.</p>	Format:	U0.16							
Format:	U0.16									
7	31:24	<b>B1L</b> <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									

<b>COLOR_PROCESSING_STATE - STD/STE State</b>						
	23:16	<b>B0L</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">133</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>P3L</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">236</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>P2L</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">236</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	26:16	<b>S0L</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">S2.8</td> </tr> </table> <p>Slope 0 of the lower part of the detection PWLF. The default is -5/256.</p>	Format:	S2.8		
Format:	S2.8					
15:8	<b>B3L</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">130</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>B2L</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">130</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
21:11	<b>S2L</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">S2.8</td> </tr> </table> <p>Slope 2 of the lower part of the detection PWLF. The default is 0/256.</p>	Format:	S2.8			
Format:	S2.8					

<b>COLOR_PROCESSING_STATE - STD/STE State</b>						
	10:0	<p><b>S1L</b></p> <table border="1"> <tr> <td>Format:</td> <td>S2.8</td> </tr> </table> <p>Slope 1 of the lower part of the detection PWLF. The default is 0/256.</p>	Format:	S2.8		
Format:	S2.8					
10	31:27	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	26:19	<p><b>P1U</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>66</td> </tr> <tr> <td>Format:</td> <td>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	<p><b>P0U</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>46</td> </tr> <tr> <td>Format:</td> <td>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	<p><b>S3L</b></p> <table border="1"> <tr> <td>Format:</td> <td>S2.8</td> </tr> </table> <p>Slope 3 of the lower part of the detection PWLF. The default is 0/256.</p>	Format:	S2.8			
Format:	S2.8					
11	31:24	<p><b>B1U</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>163</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 1 of the upper part of the detection PWLF.</p>	Default Value:	163	Format:	U8
	Default Value:	163				
	Format:	U8				
23:16	<p><b>B0U</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>143</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 0 of the upper part of the detection PWLF.</p>	Default Value:	143	Format:	U8	
Default Value:	143					
Format:	U8					
15:8	<p><b>P3U</b></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 upper part of the detection PWLF.</p>	Default Value:	236	Format:	U8	
Default Value:	236					
Format:	U8					



<b>COLOR_PROCESSING_STATE - STD/STE State</b>						
	7:0	<b>P2U</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">150</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>Y Point 2 of the upper part of the detection PWLF.</p>	Default Value:	150	Format:	U8
Default Value:	150					
Format:	U8					
12	31:27	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	26:16	<b>S0U</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">S2.8</td> </tr> </table> <p>Slope 0 of the upper part of the detection PWLF. The default is 256/256.</p>	Format:	S2.8		
Format:	S2.8					
15:8	<b>B3U</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">140</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>B2U</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">200</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
21:11	<b>S2U</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">S2.8</td> </tr> </table> <p>Slope 2 of the upper part of the detection PWLF. The default is -179/256.</p>	Format:	S2.8			
Format:	S2.8					
10:0	<b>S1U</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">S2.8</td> </tr> </table> <p>Slope 1 of the upper part of the detection PWLF. The default is -113/256.</p>	Format:	S2.8			
Format:	S2.8					
14	31:28	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					

<b>COLOR_PROCESSING_STATE - STD/STE State</b>									
	27:20	<b>Skin Types Margin</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">20</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>Skin types Y margin.</p>	Default Value:	20	Format:	U8			
	Default Value:	20							
	Format:	U8							
	19:12	<b>Skin Types Thresh</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">120</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U8</td> </tr> </table> <p>Skin types Y threshold.</p>	Default Value:	120	Format:	U8			
Default Value:	120								
Format:	U8								
11	<b>Skin Type Enable</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">Enable</td> </tr> </table> <p>Treat differently bright and dark skin types.</p> <table border="1" style="width: 100%; text-align: center;"> <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>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	<b>S3U</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">S2.8</td> </tr> </table> <p>Slope 3 of the upper part of the detection PWLF. The default is 0/256.</p>	Format:	S2.8						
Format:	S2.8								
15	31	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
	Access:	RO							
	Format:	MBZ							
	30:21	<b>SATB1</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">S7.2</td> </tr> </table> <p>First bias for the saturation PWLF (bright skin). The default is -8/4.</p>	Format:	S7.2					
	Format:	S7.2							
20:14	<b>SATP3</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">31</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">S6</td> </tr> </table> <p>Third point for the saturation PWLF (bright skin).</p>	Default Value:	31	Format:	S6				
Default Value:	31								
Format:	S6								
13:7	<b>SATP2</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">6</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">S6</td> </tr> </table> <p>Second point for the saturation PWLF (bright skin).</p>	Default Value:	6	Format:	S6				
Default Value:	6								
Format:	S6								
6:0	<b>SATP1</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">S6</td> </tr> </table> <p>First point for the saturation PWLF (bright skin). The default is -6.</p>	Format:	S6						
Format:	S6								

<b>COLOR_PROCESSING_STATE - STD/STE State</b>						
16	31	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	30:20	<p><b>SATS0</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U3.8</td> </tr> </table> <p>Zeroth slope for the saturation PWLF (bright skin). The default is 297/256.</p>	Format:	U3.8		
Format:	U3.8					
19:10	<p><b>SATB3</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>S7.2</td> </tr> </table> <p>Third bias for the saturation PWLF (bright skin). The default is 124/4.</p>	Format:	S7.2			
Format:	S7.2					
9:0	<p><b>SATB2</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>S7.2</td> </tr> </table> <p>Second bias for the saturation PWLF (bright skin). The default is 8/4.</p>	Format:	S7.2			
Format:	S7.2					
17	31:22	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
21:11	<p><b>SATS2</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U3.8</td> </tr> </table> <p>Second slope for the saturation PWLF (bright skin). The default is 297/256.</p>	Format:	U3.8			
Format:	U3.8					
10:0	<p><b>SATS1</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U3.8</td> </tr> </table> <p>First slope for the saturation PWLF (bright skin). The default is 85/256.</p>	Format:	U3.8			
Format:	U3.8					
18	31:25	<p><b>HUEP3</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>14</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> <p>Third point for the hue PWLF (bright skin)</p>	Default Value:	14	Format:	S6
	Default Value:	14				
	Format:	S6				
24:18	<p><b>HUEP2</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>6</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> <p>Second point for the hue PWLF (bright skin)</p>	Default Value:	6	Format:	S6	
Default Value:	6					
Format:	S6					
17:11	<p><b>HUEP1</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>S6</td> </tr> </table> <p>First point for the hue PWLF (bright skin). The default is -6.</p>	Format:	S6			
Format:	S6					

COLOR_PROCESSING_STATE - STD/STE State		
19	10:0	<b>SATS3</b> Format: U3.8 Third slope for the saturation PWLF (bright skin). The default is 256/256.
	31:30	<b>Reserved</b> Access: RO Format: MBZ
	29:20	<b>HUEB3</b> Format: S7.2 Third bias for the hue PWLF (bright skin). The default is 56/4.
	19:10	<b>HUEB2</b> Format: S7.2 Second bias for the hue PWLF (bright skin). The default is 8/4.
20	9:0	<b>HUEB1</b> Format: S7.2 First bias for the hue PWLF (bright skin). The default is -8/4.
	31:22	<b>Reserved</b> Access: RO Format: MBZ
	21:11	<b>HUES1</b> Format: U3.8 First slope for the hue PWLF (bright skin) The default is 85/256.
	10:0	<b>HUES0</b> Format: U3.8 Zeroth slope for the hue PWLF (bright skin) The default is 384/256.
21	31:22	<b>Reserved</b> Access: RO Format: MBZ
	21:11	<b>HUES3</b> Format: U3.8 Third slope for the hue PWLF (bright skin) The default is 256/256.

<b>COLOR_PROCESSING_STATE - STD/STE State</b>						
22	10:0	<b>HUES2</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table> <p>Second slope for the hue PWLF (bright skin) The default is 384/256.</p>	Format:	U3.8		
	Format:	U3.8				
	31	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	30:21	<b>SATB1_DARK</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S7.2</td> </tr> </table> <p>First bias for the saturation PWLF (dark skin) The default is 0/4.</p>	Format:	S7.2		
Format:	S7.2					
20:14	<b>SATP3_DARK</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>31</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> <p>Third point for the saturation PWLF (dark skin)</p>	Default Value:	31	Format:	S6	
Default Value:	31					
Format:	S6					
13:7	<b>SATP2_DARK</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>31</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> <p>Second point for the saturation PWLF (dark skin)</p>	Default Value:	31	Format:	S6	
Default Value:	31					
Format:	S6					
6:0	<b>SATP1_DARK</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S6</td> </tr> </table> <p>First point for the saturation PWLF (dark skin). The default is -11.</p>	Format:	S6			
Format:	S6					
23	31	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	30:20	<b>SATS0_DARK</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table> <p>Zeroth slope for the saturation PWLF (dark skin). The default is 397/256.</p>	Format:	U3.8		
Format:	U3.8					
19:10	<b>SATB3_DARK</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S7.2</td> </tr> </table> <p>Third bias for the saturation PWLF (dark skin). The default is 124/4.</p>	Format:	S7.2			
Format:	S7.2					
9:0	<b>SATB2_DARK</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S7.2</td> </tr> </table> <p>Second bias for the saturation PWLF (dark skin). The default is 124/4.</p>	Format:	S7.2			
Format:	S7.2					

COLOR_PROCESSING_STATE - STD/STE State					
24	31:22	<b>Reserved</b>			
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
Format:	MBZ				
21:11	<b>SATS2_DARK</b> <table border="1"> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table> Second slope for the saturation PWLF (dark skin). The default is 256/256.	Format:	U3.8		
Format:	U3.8				
25	31:25	<b>HUEP3_DARK</b>			
		<table border="1"> <tr> <td>Default Value:</td> <td>14</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> Third point for the hue PWLF (dark skin).	Default Value:	14	Format:
	Default Value:	14			
Format:	S6				
24:18	<b>HUEP2_DARK</b> <table border="1"> <tr> <td>Default Value:</td> <td>2</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> Third point for the hue PWLF (dark skin).	Default Value:	2	Format:	S6
Default Value:	2				
Format:	S6				
26	31:30	<b>Reserved</b>			
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
Format:	MBZ				
29:20	<b>HUEB3_DARK</b> <table border="1"> <tr> <td>Format:</td> <td>S7.2</td> </tr> </table> Third bias for the hue PWLF (dark skin). The default is 56/4.	Format:	S7.2		
Format:	S7.2				

COLOR_PROCESSING_STATE - STD/STE State		
	19:10	<b>HUEB2_DARK</b> Format: S7.2 Second bias for the hue PWLF (dark skin). The default is 0/4.
	9:0	<b>HUEB1_DARK</b> Format: S7.2 First bias for the hue PWLF (dark skin). The default is 0/4.
27	31:22	<b>Reserved</b> Access: RO Format: MBZ
	21:11	<b>HUES1_DARK</b> Format: U3.8 First slope for the hue PWLF (dark skin). The default is 0/256.
	10:0	<b>HUES0_DARK</b> Format: U3.8 Zeroth slope for the hue PWLF (dark skin). The default is 256/256.
28	31:22	<b>Reserved</b> Access: RO Format: MBZ
	21:11	<b>HUES3_DARK</b> Format: U3.8 Third slope for the hue PWLF (dark skin). The default is 256/256.
	10:0	<b>HUES2_DARK</b> 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			
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	<b>SatFactor3</b>	
		Default Value:	220
		Format:	U1.7
	The saturation factor for yellow.		
	23:16	<b>SatFactor2</b>	
Default Value:		220	
Format:		U1.7	
The saturation factor for red.			
15:8	<b>SatFactor1</b>		
	Default Value:	220	
	Format:	U1.7	
The saturation factor for magenta.			
7	<b>TCC Enable</b>		
Format:	Enable		
6:0	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
1	31:24	<b>SatFactor6</b>	
		Default Value:	220
		Format:	U1.7
	The saturation factor for blue.		
	23:16	<b>SatFactor5</b>	
Default Value:		220	
Format:		U1.7	
The saturation factor for cyan.			



COLOR_PROCESSING_STATE - TCC State						
	15:8	<b>SatFactor4</b> <table border="1"> <tr> <td>Default Value:</td> <td>220</td> </tr> <tr> <td>Format:</td> <td>U1.7</td> </tr> </table> <p>The saturation factor for green.</p>	Default Value:	220	Format:	U1.7
	Default Value:	220				
Format:	U1.7					
7:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
2	31:30	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	29:20	<b>Base Color 3</b> <table border="1"> <tr> <td>Default Value:</td> <td>483</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table>	Default Value:	483	Format:	U10
Default Value:	483					
Format:	U10					
19:10	<b>Base Color 2</b> <table border="1"> <tr> <td>Default Value:</td> <td>307</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table>	Default Value:	307	Format:	U10	
Default Value:	307					
Format:	U10					
9:0	<b>Base Color 1</b> <table border="1"> <tr> <td>Default Value:</td> <td>145</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table>	Default Value:	145	Format:	U10	
Default Value:	145					
Format:	U10					
3	31:30	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	29:20	<b>Base Color 6</b> <table border="1"> <tr> <td>Default Value:</td> <td>995</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table>	Default Value:	995	Format:	U10
Default Value:	995					
Format:	U10					
19:10	<b>Base Color 5</b> <table border="1"> <tr> <td>Default Value:</td> <td>819</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table>	Default Value:	819	Format:	U10	
Default Value:	819					
Format:	U10					
9:0	<b>Base Color 4</b> <table border="1"> <tr> <td>Default Value:</td> <td>657</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table>	Default Value:	657	Format:	U10	
Default Value:	657					
Format:	U10					
4	31:16	<b>Color Transit Slope 23</b> <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 <math>1 / (BC3 - BC2)</math> [1/62]</p>	Default Value:	744	Format:	U0.16
		Default Value:	744			
Format:	U0.16					

<b>COLOR_PROCESSING_STATE - TCC State</b>		
	15:0	<b>Color Transit Slope 12</b>
		Default Value: 405
		Format: U0.16 The calculation result of $1 / (BC2 - BC1)$ [1/57]
5	31:16	<b>Color Transit Slope 45</b>
		Default Value: 407
		Format: U0.16 The calculation result of $1 / (BC5 - BC4)$ [1/57]
	15:0	<b>Color Transit Slope 34</b>
		Default Value: 1131
		Format: U0.16 The calculation result of $1 / (BC4 - BC3)$ [1/61]
6	31:16	<b>Color Transit Slope 61</b>
		Default Value: 377
		Format: U0.16 The calculation result of $1 / (BC1 - BC6)$ [1/62]
	15:0	<b>Color Transit Slope 56</b>
		Default Value: 372
		Format: U0.16 The calculation result of $1 / (BC6 - BC5)$ [1/62]
7	31:22	<b>Color Bias 3</b>
		Default Value: 0
		Format: U2.8 Color bias for BaseColor3.
	21:12	<b>Color Bias 2</b>
		Default Value: 150
		Format: U2.8 Color bias for BaseColor2.
	11:2	<b>Color Bias 1</b>
		Default Value: 0
		Format: U2.8 Color bias for BaseColor1.

COLOR_PROCESSING_STATE - TCC State						
	1:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					
8	31:22	<b>Color Bias 6</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U2.8</td> </tr> </table> Color bias for BaseColor6.	Default Value:	0	Format:	U2.8
		Default Value:	0			
	Format:	U2.8				
	21:12	<b>Color Bias 5</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U2.8</td> </tr> </table> Color bias for BaseColor5.	Default Value:	0	Format:	U2.8
Default Value:		0				
Format:	U2.8					
11:2	<b>ColorBias4</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U2.8</td> </tr> </table> Color bias for BaseColor4.	Default Value:	0	Format:	U2.8	
	Default Value:	0				
Format:	U2.8					
1:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO				
Format:	MBZ					
9	31	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
		Access:	RO			
	Format:	MBZ				
	30:24	<b>UV Threshold</b> <table border="1"> <tr> <td>Default Value:</td> <td>3</td> </tr> <tr> <td>Format:</td> <td>U7</td> </tr> </table> Low UV threshold.	Default Value:	3	Format:	U7
Default Value:		3				
Format:	U7					
23:19	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO				
Format:	MBZ					
18:16	<b>UV Threshold Bits</b> <table border="1"> <tr> <td>Default Value:</td> <td>3</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> Low UV transition width bits.	Default Value:	3	Format:	U3	
	Default Value:	3				
Format:	U3					

COLOR_PROCESSING_STATE - TCC State			
	15:13	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	12:8	<b>STE Threshold</b>	
		Default Value: 0	
		Format: U5	
	Skin tone pixels enhancement threshold.		
	7:3	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	2:0	<b>STE Slope Bits</b>	
		Default Value: 0	
Format: U3			
Skin tone pixels enhancement slope bits.			
10	31:16	<b>Inverse UVMax Color</b>	
		Default Value: 146	
		Format: U0.16	
	1 / UVMaxColor. Used for the SFs2 calculation.		
	15:9	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	8:0	<b>UVMax Color</b>	
		Default Value: 448	
Format: U9			
The maximum absolute value of the legal UV pixels. Used for the SFs2 calculation.			

## Color Calculator State Pointer Message Header Control

<b>MHC_RT_CCSP - Color Calculator State Pointer Message Header Control</b>					
Size (in bits):		32			
Default Value:		0x00000000			
DWord	Bit	Description			
0	31:6	<p><b>Color Calculator State Pointer</b></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><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				



## Color Code Message Header Control

MHC_RT_CC - Color Code Message Header Control			
Size (in bits):		32	
Default Value:		0x00000000	
DWord	Bit	Description	
0	31:10	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	9:8	<b>Color Code</b>	
		Format:	U2
			This ID is assigned by the Windower unit and is used to track synchronizing events. Reserved for HW implementation use
7:0	<b>FTID</b>		
	Format:	U8	
		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.	

## COMPRESSION\_PAIR\_BIT

COMPRESSION_PAIR_BIT														
Size (in bits):	2													
Default Value:	0x00000000													
DWord	Bit	Description												
0	1:0	<p><b>COMPRESSION PAIRING BIT</b> This field defines which 2 cachelines are combined in a 128B memory compression block.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Bit_6 [Default]</td> <td>Pairing bit is Addr[6]: Two consecutive cachelines form a 128B compression block.</td> </tr> <tr> <td>1h</td> <td>Bit_7</td> <td>Pairing bit is Addr[7]: Two cachelines with a 128B stride form a 128B compression block.</td> </tr> <tr> <td>2h</td> <td>Bit_8</td> <td>Pairing bit is Addr[8]: Two cachelines with a 256B stride form a 128B compression block.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Bit_6 [Default]	Pairing bit is Addr[6]: Two consecutive cachelines form a 128B compression block.	1h	Bit_7	Pairing bit is Addr[7]: Two cachelines with a 128B stride form a 128B compression block.	2h	Bit_8	Pairing bit is Addr[8]: Two cachelines with a 256B stride form a 128B compression block.
Value	Name	Description												
0h	Bit_6 [Default]	Pairing bit is Addr[6]: Two consecutive cachelines form a 128B compression block.												
1h	Bit_7	Pairing bit is Addr[7]: Two cachelines with a 128B stride form a 128B compression block.												
2h	Bit_8	Pairing bit is Addr[8]: Two cachelines with a 256B stride form a 128B compression block.												



## ComputeCS Hardware-Detected Error Bit Definitions

ComputeCS Hardware-Detected Error Bit Definitions								
Source:	ComputeCS							
Size (in bits):	32							
Default Value:	0x00000000							
DWord	Bit	Description						
0	31:12	<b>Reserved</b>						
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
	Access:	RO						
	Format:	MBZ						
	11	<b>Reserved</b>						
<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ			
Access:	RO							
Format:	MBZ							
10:3	<b>Reserved</b>							
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO							
Format:	MBZ							
2	<b>Command Privilege Violation Error</b> 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	<b>Reserved</b>							
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO							
Format:	MBZ							
0	<b>Instruction Error</b> 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"> <li>Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported).</li> <li>Defeatured MI Instruction Opcodes:</li> </ul>							
	<table border="1"> <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>		Value	Name	Description	1		Instruction Error detected
	Value	Name	Description					
	1		Instruction Error detected					
<p style="text-align: center;"><b>Programming Notes</b></p>								
This error indications cannot be cleared except by reset (i.e., it is a fatal error).								



## Compute Engine Interrupt Vector

COMPUTE_INTR_VEC - Compute Engine Interrupt Vector						
Size (in bits):	16					
Default Value:	0x00000000					
DWord	Bit	Description				
0	15	<p><b>Catastrophic Error</b></p> <p>This interrupt signals that a unrecoverable error (for e.g encountered fault when accessing a page mapped in Global GTT) during the engine processing. When Memory interface signals this error, the Command Streamer will stop parsing any more instructions. Scheduler is expected to reset the engine to evict the context</p>				
	14	<p><b>EU Restart Interrupt</b></p> <p>EU Restart Interrupt is generated by the GA fabric, and not by Render Command Streamer. GA routes this interrupt to GuC independently of Command Stream.</p>				
	13	<p><b>Context Stall</b></p> <p>Command streamer will generate a Context Stall interrupt when a high priority context gets stalled due to the other command streamer executing a normal priority or low priority context is "Run Alone" mode OR Command streamer will generate a Context Stall interrupt when a high priority context gets stalled while procuring run alone mode.</p>				
	12	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	11	<b>CS Wait On Semaphore</b>				
	10	<b>Spare 10</b>				
	9	<b>CS TR Invalid Tile Detection</b>				
	8	<b>CS Context Switch Interrupt</b>				
	7	<p><b>Page Fault Interrupt</b></p> <p>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.</p> <p>In Advanced (PRQ) Fault Interface is done through GUC interface.</p>				
	6	<b>CS Watchdog Counter Expired</b>				
	5	<b>Spare 5</b>				
	4	<b>CS PIPE_CONTROL Notify</b>				
	3	<b>CS Error Interrupt</b>				
	2:1	<b>Spare 2</b>				



<b>COMPUTE_INTR_VEC - Compute Engine Interrupt Vector</b>	
---	--

0	<b>CS MI User Interrupt</b>
---	-----------------------------

## Context Descriptor Format

<b>CONTEXT_DESCRIPTOR - Context Descriptor Format</b>	
Size (in bits):	64
Default Value:	0x00000000, 0x00000000
Description	
<p>This is the format of context descriptors which make up submitted execlists.</p> <p>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.</p> <ul style="list-style-type: none"> <li>Context ID is used for semaphore signaling by hardware and software.</li> <li>Context ID matching is used by hardware to detect Lite Restore.</li> <li>Context ID is used by hardware for page fault reporting and response with IOMMU.</li> <li>Context switch reason and the associated Context ID are reported to Context Switch Status Buffer by hardware on a context switch.</li> </ul>	
<p>Context ID which is a 32 bit field is further divided in to following segments described below:</p> <ul style="list-style-type: none"> <li><b>Bits[63:61] (Bits 31:29 of Context ID)</b> represents <b>Engine class</b>.</li> <li><b>Bits[60:55] (Bits 28:23 of Context ID)</b> represents <b>SW Counter</b>.</li> <li><b>Bit[54] (Bit 22 of Context ID)</b> MBZ for SW programming; this bit is used by hardware to distinguish between F&amp;H vs F&amp;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.</li> <li><b>Bits[53:48] (Bits 21:16 of Context ID)</b> represents <b>Engine Instance</b> (within a Engine class).</li> <li><b>Bits[47:37] (Bits 15:5 of Context ID)</b> represents <b>SW Context ID</b> which is a software assigned unique context ID. (supports 2048 contexts per virtual function)</li> <li><b>Bits[36:32] (Bits 4:0 of Context ID)</b> represents <b>Virtual Function Number</b> (when virtualization is enabled). Set to zero when virtualization is not enabled. This field contains the bits [4:0] of the Virtual Function Number.</li> </ul> <p><b>Programming Note:</b> "Virtual Function Number" must be always programmed to value 0x0.</p> <p>Hardware compares the following fields of the outgoing context to that of the incoming context to detect a lite restore. Lite restore is detected when the following fields are equal and the incoming context does not have the "Force Restore" bit set. On a lite restore hardware will only sample the tail pointer from memory (LRCA) and keep executing the ongoing context with out initiating any context switch flows (Flush, Context Save, Context Restore). Lite restore is HW detected context switch optimization transparent to SW, Context Switch Status report and Context Switch Interrupt generation happens on a lite restore, Hardware Front End may temporarily get stalled from parsing new commands.</p> <ul style="list-style-type: none"> <li><b>DW1.SW Context ID</b></li> <li><b>DW1.Virtual Function Number</b></li> <li><b>DW0.Logical Ring Context Address (LRCA)</b></li> <li><b>DW0. Reserved Bits[11:9]</b></li> </ul>	
Context ID issued for comparing during lite restore and context specific OA enabling.	

## CONTEXT\_DESCRIPTOR - Context Descriptor Format

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.

DWord	Bit	Description							
0..1	63:61	<b>Context ID - Engine Class</b> Bits 31:29 of Context ID represents Engine class.							
	60:55	<b>Context ID - SW Counter</b> Bits 28:23 of Context ID represents SW Counter.							
	54	<b>Reserved</b> Bits 22 of Context ID. This must be zero 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.							
	53:48	<b>Context ID - Engine Instance</b> Bits 21:16 of Context ID represents Engine Instance.							
	47:37	<b>Context ID - SW Context ID</b> Bits 15:5 of Context ID represents SW Context ID, which is a software assigned unique context ID. (supports 2048 contexts per virtual function).							
	36:32	<b>Context ID - Virtual Function Number</b> Bits 4:0 of Context ID represents the Virtual Function Number (when virtualization is enabled). Set to zero when virtualization is not enabled. This field contains the bits [4:0] of the Virtual Function Number.							
	31:12	<b>Logical Ring Context Address (LRCA)</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[31:12]</td> </tr> </table> This field contains the 4 KB-aligned address of the Logical Ring Context associated with his exelist element. LRCA must be always programmed in GGTT memory.	Format:	GraphicsAddress[31:12]					
	Format:	GraphicsAddress[31:12]							
11	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO								
Format:	MBZ								
10:9	<b>Context Priority</b> This field indicates the prioritization of the thread dispatch associated with the corresponding context. Note that Render Engine and Compute Engine are executing contexts of their own with the corresponding priority programmed. For e.g: When Compute Engine is executing lower priority context when compared to the context executed by render engine, then threads dispatched from render engine (3D - VS, HS, DS, GS & PSD and GPGPU -TSG threads corresponding to render engine) are given priority over the TSG threads dispatched for compute engine. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Low Priority</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Normal Priority</td> </tr> <tr> <td style="text-align: center;">2</td> <td>High Priority</td> </tr> </tbody> </table>	Value	Name	0	Low Priority	1	Normal Priority	2	High Priority
Value	Name								
0	Low Priority								
1	Normal Priority								
2	High Priority								

## CONTEXT\_DESCRIPTOR - Context Descriptor Format

		<b>Programming Notes</b>	
		This field is only functional for RenderCS and ComputeCS and must be only programmed for context descriptor submitted to RenderCS and ComputeCS.	
8	<b>Privilege Access</b>		
	<b>Description</b>		
	This field when set indicates PPGTT enabled.		
	<b>Programming Notes</b>		
	This field must be always set.		
7:6	<b>Fault Handling</b>		
	Source:	CommandStreammer	
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	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. CommandStreammer will not initiate context switch on occurrence of Fault Error.
	Others	Reserved	Reserved
	<b>Programming Notes</b>		
	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 for the corresponding generation.		
5	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
4:3	<b>Addressing Mode &amp; Legacy Context</b>		
	Format:	U2	
	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. GFX engine always uses 32b virtual addressing mode when translated using GGTT irrespective of below options.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	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.

## CONTEXT\_DESCRIPTOR - Context Descriptor Format

	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.				
		Others	Reserved	Other values are not supported.			
	2	<p><b>Force Restore</b></p> <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 <b>Render Context Restore Inhibit</b> 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	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>			Access:	RO	Format:
Access:	RO						
Format:	MBZ						
0	<p><b>Valid</b></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							
Size (in bits):	64						
Default Value:	0x00000000, 0x00000000						
<p>The context status is an update sent by a Command Streamer to the scheduler.            The Engine Class and Instance ID specifies the Command streamer the event came from.            See the <b>Engine ID Definition</b> structure.            Context ID Away: Context ID of the context that the command streamer is switching away from.            Context ID To: Context ID of the context that the command streamer is switching execution to</p>							
DWord	Bit	Description					
0	31:26	<b>Context ID To SW Counter</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U6</td></tr></table>		U6			
		U6					
	25:15	<b>Context ID To SW Context ID</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U11</td></tr></table> 0x7FF: Is reserved to indicate HW idle state. "Ctxt-ID To SW Context-ID" set to 0x7FF in the report indicates HW went to Idle following this context switch. Indicate Active to Idle switch		U11			
		U11					
	14:12	<b>Context ID To Virtual Function Number</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U3</td></tr></table> This field contains the bits [2:0] of the Virtual Function Number.		U3			
		U3					
	11:6	<b>Engine Instance</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U6</td></tr></table>		U6			
		U6					
	5:3	<b>Engine Class</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U3</td></tr></table>		U3			
		U3					
2	<b>Semaphore Wait Mode</b> This field indicates the Semaphore Wait Mode (Poll or Signal) when the context switch is due to "Wait on Semaphore". This field is only valid when "Switch Detail" indicates "Wait on Semaphore". <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Signal Mode</td> </tr> <tr> <td>1</td> <td>Poll Mode</td> </tr> </tbody> </table>	Value	Name	0	Signal Mode	1	Poll Mode
Value	Name						
0	Signal Mode						
1	Poll Mode						
1	<b>Reserved</b> Access: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>RO</td></tr></table> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>MBZ</td></tr></table>		RO		MBZ		
	RO						
	MBZ						
0	<b>Switched to New Queue</b> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>False</td> </tr> <tr> <td>1</td> <td>True</td> </tr> </tbody> </table>	Value	Name	0	False	1	True
Value	Name						
0	False						
1	True						

<b>Context Status</b>														
1	31:26	<b>Context ID Away SW Counter</b> Format: <span style="float: right;">U6</span>												
	25:15	<b>Context ID Away SW Context ID</b> Format: <span style="float: right;">U11</span> 0x7FF: Is reserved to indicate HW idle state. Ctxt-ID Away SW Context-ID set to 0x7FF in the CSB report indicates HW was Idle with no valid context at the time of context switch. Indicates Idle to Active switch.												
	14:12	<b>Context ID Away Virtual Function Number</b> Format: <span style="float: right;">U3</span> This field contains the bits [2:0] of the Virtual Function Number.												
	11:5	<b>Wait Detail</b> Format: <span style="float: right;">U7</span> This field is only valid when Switch Detail" indicates "Wait on Sync Flip" or "Wait on Scanline" or "Wait on VBlank" or "Wait on Semaphore". <ul style="list-style-type: none"> <li>This field indicates the Display Plane ID when the "Switch Detail" indicates "Wait on Sync Flip".</li> <li>This field indicates the Display PipeID when the "Switch Detail" indicates "Wait on Scanline" or "Wait on VBlank".</li> <li>This field indicates the Wait Token Number when the "Switch Detail" indicates "Wait on Semaphore".</li> </ul> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>[0,27]</td> <td>Wait on Display</td> <td>The value entered here is the Display Plane ID.</td> </tr> <tr> <td>[0,26]</td> <td>Wait on Semaphore</td> <td>The value entered here is the Wait Token Number.</td> </tr> </tbody> </table>	Value	Name	Description	[0,27]	Wait on Display	The value entered here is the Display Plane ID.	[0,26]	Wait on Semaphore	The value entered here is the Wait Token Number.			
Value	Name	Description												
[0,27]	Wait on Display	The value entered here is the Display Plane ID.												
[0,26]	Wait on Semaphore	The value entered here is the Wait Token Number.												
4	<b>Reserved</b> Access: <span style="float: right;">RO</span> Format: <span style="float: right;">MBZ</span>													
3:0	<b>Switch Detail</b> Format: <span style="float: right;">U4</span> Any values not listed below are reserved. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Context Complete</td> <td>Indicates context is complete with Head Pointer equal to Tail Pointer. In case of Render Engine it implies both RenderCS and PositionCS are complete with head pointer equal to tail pointer.</td> </tr> <tr> <td>1</td> <td>Wait on Sync Flip</td> <td></td> </tr> <tr> <td>2</td> <td>Wait on VBlank</td> <td></td> </tr> </tbody> </table>		Value	Name	Description	0	Context Complete	Indicates context is complete with Head Pointer equal to Tail Pointer. In case of Render Engine it implies both RenderCS and PositionCS are complete with head pointer equal to tail pointer.	1	Wait on Sync Flip		2	Wait on VBlank	
Value	Name	Description												
0	Context Complete	Indicates context is complete with Head Pointer equal to Tail Pointer. In case of Render Engine it implies both RenderCS and PositionCS are complete with head pointer equal to tail pointer.												
1	Wait on Sync Flip													
2	Wait on VBlank													



Context Status			
		3	Wait on Scanline
		4	Wait on Semaphore
		5	Context Pre-empted This field is set when the context is preempted on a preemptable command and the command is not MI_SEMAPHORE_WAIT or MI_WAIT_FOR_EVENT.
		[6h-Fh]	Reserved

## CPS\_STATE

<b>CPS_STATE</b>																
Source:	RenderCS															
Size (in bits):	256															
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000															
DWord	Bit	Description														
0	31:27	<b>Reserved</b>														
	26:16	<b>MinCPSizeY</b>														
		Format: S3.7 This bit-field defines the minimum shading ratio in Y dimension in screen space. This value is used only when Coarse Pixel Shading is enabled. It also defines the floor of the non-quantized CPSizeY for CPS_MODE_RADIAL. HW quantizes this value to determine Decoupled Rate. This value is used to clamp the CPSizeY for the lowest bound.														
	15	<b>RESERVED</b>														
	14	<b>ScaleAxis</b>														
		Format: U1 This bit defines which dimension (along X- or Y- axis) should be scaled when computing Coarse Pixel Size values along ellipse in CPS_MODE_RADIAL.														
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>X axis</td> <td>Use aspect to scale X-dimension</td> </tr> <tr> <td>1h</td> <td>Y axis</td> <td>Use aspect to scale Y-dimension</td> </tr> </tbody> </table>	Value	Name	Description	0h	X axis	Use aspect to scale X-dimension	1h	Y axis	Use aspect to scale Y-dimension					
		Value	Name	Description												
	0h	X axis	Use aspect to scale X-dimension													
	1h	Y axis	Use aspect to scale Y-dimension													
13:12	<b>Coarse Pixel Shading Mode</b>															
	Format: U2 This bit-field defines Coarse Pixel Shading Mode.															
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>CPS_MODE_NONE</td> <td>Coarse Pixel Shading is disabled. HW may be required to drive default values to shader inputs e.g. ScaleX = ScaleY = 1 and LODCompX = LODCompY = 1.</td> </tr> <tr> <td>1h</td> <td>CPS_MODE_CONSTANT</td> <td>Coarse Pixel Shading Ratios are defined per DRAW based on MinCPSizeX and MinCPSizeY fields in this state (constant across render target).</td> </tr> <tr> <td>2h</td> <td>CPS_MODE_RADIAL</td> <td>Coarse Pixel Shading Ratio varies radially from a focal point defined by (X_Focal, Y_Focal) relative to the viewport X/Y origin. This mode is typically used when there is Depth of Field or Ring of Confusion camera effects are desired.</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0h	CPS_MODE_NONE	Coarse Pixel Shading is disabled. HW may be required to drive default values to shader inputs e.g. ScaleX = ScaleY = 1 and LODCompX = LODCompY = 1.	1h	CPS_MODE_CONSTANT	Coarse Pixel Shading Ratios are defined per DRAW based on MinCPSizeX and MinCPSizeY fields in this state (constant across render target).	2h	CPS_MODE_RADIAL	Coarse Pixel Shading Ratio varies radially from a focal point defined by (X_Focal, Y_Focal) relative to the viewport X/Y origin. This mode is typically used when there is Depth of Field or Ring of Confusion camera effects are desired.	3h	Reserved	
	Value	Name	Description													
	0h	CPS_MODE_NONE	Coarse Pixel Shading is disabled. HW may be required to drive default values to shader inputs e.g. ScaleX = ScaleY = 1 and LODCompX = LODCompY = 1.													
	1h	CPS_MODE_CONSTANT	Coarse Pixel Shading Ratios are defined per DRAW based on MinCPSizeX and MinCPSizeY fields in this state (constant across render target).													
2h	CPS_MODE_RADIAL	Coarse Pixel Shading Ratio varies radially from a focal point defined by (X_Focal, Y_Focal) relative to the viewport X/Y origin. This mode is typically used when there is Depth of Field or Ring of Confusion camera effects are desired.														
3h	Reserved															

## CPS\_STATE

		Programming Notes				
		<p>It is a valid configuration to set the CPS mode other than CPS_MODE_NONE and request per-pixel dispatch in 3DSTATE_PS_EXTRA. In such case, 3DSTATE_PS_EXTRA configuration overrides 3DSTATE_CPS configuration, and effective CPS mode is set to CPS_MODE_NONE for this draw primitive.</p> <p>It is an INVALID configuration to set the CPS mode other than CPS_MODE_NONE and request per-sample dispatch in 3DSTATE_PS_EXTRA. Such configuration should be disallowed at the API level, and rendering results are undefined.</p> <p>It is a valid configuration to set the CPS mode to CPS_MODE_NONE and at the same time set Pixel Shader Is Per Coarse Pixel in 3DSTATE_PS_EXTRA. In such case, 3DSTATE_PS_EXTRA bit is ignored and shader is dispatched at pixel-rate; shader inputs specific to coarse-rate have undefined value (ActualCoarsePixelSize for example).</p>				
	11	<p><b>Statistics Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table>	Format:	Enable		
Format:	Enable					
	10:0	<p><b>MinCPSizeX</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>S3.7</td> </tr> </table> <p>This bit-field defines the minimum shading ratio in X dimension in screen space. This value is used only when Coarse Pixel Shading is enabled. It also defines the floor of the non-quantized ScaleX for Mode 1. HW quantizes this value to determine Decoupled Rate.</p>	Format:	S3.7		
Format:	S3.7					
1	31:27	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					
	26:16	<p><b>MaxCPSizeY</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>S3.7</td> </tr> </table> <p>This bit-field defines the maximum shading ratio in Y dimension in screen space. This value is used only when Coarse Pixel Shading is enabled and Coarse Pixel Shading Mode is set to CPS_MODE_RADIAL. This value is used to clamp the CPSizeY for the highest bound. MaxCPSizeY must be greater than or equal to MinCPSizeY when this value is used.</p>	Format:	S3.7		
Format:	S3.7					
	15:11	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					
	10:0	<p><b>MaxCPSizeX</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>S3.7</td> </tr> </table> <p>This bit-field defines the maximum shading ratio in X dimension in screen space. This value is used only when Coarse Pixel Shading is enabled and Coarse Pixel Shading Mode is set to CPS_MODE_RADIAL. This value is used to clamp the CPSizeX for the highest bound. MaxCPSizeX must be greater than or equal to MinCPSizeX when this value is used.</p>	Format:	S3.7		
Format:	S3.7					
2	31:16	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					

<b>CPS_STATE</b>																																	
	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">15:0</td> <td style="width: 10%;"><b>Y_Focal</b></td> <td style="width: 50%;"></td> <td style="width: 20%; text-align: center;">S15</td> </tr> <tr> <td colspan="4">           Format:         </td> </tr> <tr> <td colspan="4">           This field defines the Y-coordinate for a focal point with respect to which shading ratio is computed in CPS_MODE_RADIAL.         </td> </tr> <tr> <td colspan="4" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="4">           The valid data range is (-2<sup>14</sup> to 2<sup>14</sup>-1)         </td> </tr> </table>	15:0	<b>Y_Focal</b>		S15	Format:				This field defines the Y-coordinate for a focal point with respect to which shading ratio is computed in CPS_MODE_RADIAL.				<b>Programming Notes</b>				The valid data range is (-2 <sup>14</sup> to 2 <sup>14</sup> -1)															
15:0	<b>Y_Focal</b>		S15																														
Format:																																	
This field defines the Y-coordinate for a focal point with respect to which shading ratio is computed in CPS_MODE_RADIAL.																																	
<b>Programming Notes</b>																																	
The valid data range is (-2 <sup>14</sup> to 2 <sup>14</sup> -1)																																	
3	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">31:16</td> <td style="width: 10%;"><b>Reserved</b></td> <td style="width: 50%;"></td> <td style="width: 20%;"></td> </tr> <tr> <td colspan="2">Access:</td> <td colspan="2" style="text-align: center;">RO</td> </tr> <tr> <td colspan="2">Format:</td> <td colspan="2" style="text-align: center;">MBZ</td> </tr> </table> <table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">15:0</td> <td style="width: 10%;"><b>X_Focal</b></td> <td style="width: 50%;"></td> <td style="width: 20%; text-align: center;">S15</td> </tr> <tr> <td colspan="4">           Format:         </td> </tr> <tr> <td colspan="4">           This field defines the X-coordinate for a focal point with respect to which shading ratio is computed in CPS_MODE_RADIAL.         </td> </tr> <tr> <td colspan="4" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="4">           The valid data range is (-2<sup>14</sup> to 2<sup>14</sup>-1)         </td> </tr> </table>	31:16	<b>Reserved</b>			Access:		RO		Format:		MBZ		15:0	<b>X_Focal</b>		S15	Format:				This field defines the X-coordinate for a focal point with respect to which shading ratio is computed in CPS_MODE_RADIAL.				<b>Programming Notes</b>				The valid data range is (-2 <sup>14</sup> to 2 <sup>14</sup> -1)			
31:16	<b>Reserved</b>																																
Access:		RO																															
Format:		MBZ																															
15:0	<b>X_Focal</b>		S15																														
Format:																																	
This field defines the X-coordinate for a focal point with respect to which shading ratio is computed in CPS_MODE_RADIAL.																																	
<b>Programming Notes</b>																																	
The valid data range is (-2 <sup>14</sup> to 2 <sup>14</sup> -1)																																	
4	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">31:0</td> <td style="width: 10%;"><b>My</b></td> <td style="width: 50%;"></td> <td style="width: 20%; text-align: center;">IEEE_FLOAT32</td> </tr> <tr> <td colspan="4">           Format:         </td> </tr> <tr> <td colspan="4">           This field defines the slope of the transfer function for computing CPSIZEY for CPS_MODE_RADIAL.         </td> </tr> <tr> <td colspan="4" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="4">           SW needs to compute this from API supplied parameters:         </td> </tr> <tr> <td colspan="4" style="text-align: center;"> <math display="block">(M_x, M_y) = \left( \frac{S_x^{\max} - S_x^{\min}}{R_{\max} - R_{\min}}, \frac{S_y^{\max} - S_y^{\min}}{R_{\max} - R_{\min}} \right)</math> </td> </tr> <tr> <td colspan="4">           My must be greater than or equal to zero         </td> </tr> </table>	31:0	<b>My</b>		IEEE_FLOAT32	Format:				This field defines the slope of the transfer function for computing CPSIZEY for CPS_MODE_RADIAL.				<b>Programming Notes</b>				SW needs to compute this from API supplied parameters:				$(M_x, M_y) = \left( \frac{S_x^{\max} - S_x^{\min}}{R_{\max} - R_{\min}}, \frac{S_y^{\max} - S_y^{\min}}{R_{\max} - R_{\min}} \right)$				My must be greater than or equal to zero							
31:0	<b>My</b>		IEEE_FLOAT32																														
Format:																																	
This field defines the slope of the transfer function for computing CPSIZEY for CPS_MODE_RADIAL.																																	
<b>Programming Notes</b>																																	
SW needs to compute this from API supplied parameters:																																	
$(M_x, M_y) = \left( \frac{S_x^{\max} - S_x^{\min}}{R_{\max} - R_{\min}}, \frac{S_y^{\max} - S_y^{\min}}{R_{\max} - R_{\min}} \right)$																																	
My must be greater than or equal to zero																																	
5	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">31:0</td> <td style="width: 10%;"><b>Mx</b></td> <td style="width: 50%;"></td> <td style="width: 20%; text-align: center;">IEEE_FLOAT32</td> </tr> <tr> <td colspan="4">           Format:         </td> </tr> <tr> <td colspan="4">           This field defines the slope of the transfer function for computing CPSIZEX for CPS_MODE_RADIAL.         </td> </tr> <tr> <td colspan="4" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="4">           SW needs to compute this from API supplied parameters:         </td> </tr> <tr> <td colspan="4" style="text-align: center;"> <math display="block">(M_x, M_y) = \left( \frac{S_x^{\max} - S_x^{\min}}{R_{\max} - R_{\min}}, \frac{S_y^{\max} - S_y^{\min}}{R_{\max} - R_{\min}} \right)</math> </td> </tr> <tr> <td colspan="4">           Mx must be greater than or equal to zero         </td> </tr> </table>	31:0	<b>Mx</b>		IEEE_FLOAT32	Format:				This field defines the slope of the transfer function for computing CPSIZEX for CPS_MODE_RADIAL.				<b>Programming Notes</b>				SW needs to compute this from API supplied parameters:				$(M_x, M_y) = \left( \frac{S_x^{\max} - S_x^{\min}}{R_{\max} - R_{\min}}, \frac{S_y^{\max} - S_y^{\min}}{R_{\max} - R_{\min}} \right)$				Mx must be greater than or equal to zero							
31:0	<b>Mx</b>		IEEE_FLOAT32																														
Format:																																	
This field defines the slope of the transfer function for computing CPSIZEX for CPS_MODE_RADIAL.																																	
<b>Programming Notes</b>																																	
SW needs to compute this from API supplied parameters:																																	
$(M_x, M_y) = \left( \frac{S_x^{\max} - S_x^{\min}}{R_{\max} - R_{\min}}, \frac{S_y^{\max} - S_y^{\min}}{R_{\max} - R_{\min}} \right)$																																	
Mx must be greater than or equal to zero																																	
6	<table border="1" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">31:0</td> <td style="width: 10%;"><b>Rmin</b></td> <td style="width: 50%;"></td> <td style="width: 20%; text-align: center;">IEEE_FLOAT32</td> </tr> <tr> <td colspan="4">           Format:         </td> </tr> <tr> <td colspan="4">           This field defines (smaller) radius of the inner ellipse for CPS_MODE_RADIAL. All points on inner ellipse have coarse point size = (MinCPSIZEX, MinCPSIZEY).         </td> </tr> </table>	31:0	<b>Rmin</b>		IEEE_FLOAT32	Format:				This field defines (smaller) radius of the inner ellipse for CPS_MODE_RADIAL. All points on inner ellipse have coarse point size = (MinCPSIZEX, MinCPSIZEY).																							
31:0	<b>Rmin</b>		IEEE_FLOAT32																														
Format:																																	
This field defines (smaller) radius of the inner ellipse for CPS_MODE_RADIAL. All points on inner ellipse have coarse point size = (MinCPSIZEX, MinCPSIZEY).																																	

<b>CPS_STATE</b>				
7	31:0	<p><b>Aspect</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT32</td> </tr> </table> <p>This field defines aspect for both inner and outer ellipses in CPS_MODE_RADIAL. The aspect parameter must be within &lt;0,1&gt; range and Driver must program it as ratio of smallest ellipse radius to larger ellipse radius: <math>Aspect = \min(radiusX, radiusY) / \max(radiusX, radiusY)</math> where radiusX and radiusY define ellipse radius along x- and y- axes respectively. Note: Aspect must be same for both inner and outer ellipses.</p>	Format:	IEEE_FLOAT32
Format:	IEEE_FLOAT32			



## CSC COEFFICIENT FORMAT

CSC COEFFICIENT FORMAT																										
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	<b>Sign</b>																								
		<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	<b>Exponent_bits</b> 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	<b>Mantissa</b>																									
2:0	<b>Reserved</b>																									
	Access:	RO																								
	Format:	MBZ																								

## Data Port 0 Message Types

MT_DP0 - Data Port 0 Message Types																																
Source:	EuSubFunctionDataPort0																															
Size (in bits):	5																															
Default Value:	0x00000000																															
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.																																
DWord	Bit	Description																														
0	4	<b>Legacy DAP-DC Message</b> 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	<b>Message Type</b> 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>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>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	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	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																													
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																														
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:	EuSubFunctionDataPort1																																																																												
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	<p><b>Message Type</b> Specifies type of message</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>01h</td><td>MT1R_US</td><td>Untyped Surface Read message</td></tr> <tr><td>02h</td><td>MT1A_UI</td><td>Untyped Atomic Integer Operation message</td></tr> <tr><td>03h</td><td>MT1A_UHI</td><td>Untyped Atomic Half Integer Operation message</td></tr> <tr><td>04h</td><td>MT1R_MB</td><td>Media Block Read message</td></tr> <tr><td>05h</td><td>MT1R_TS</td><td>Typed Surface Read message</td></tr> <tr><td>06h</td><td>MT1A_TA</td><td>Typed Atomic Integer Operation message</td></tr> <tr><td>07h</td><td>MT1A_TAH</td><td>Typed Atomic Half Integer Operation message</td></tr> <tr><td>08h</td><td>Reserved</td><td>Ignored</td></tr> <tr><td>09h</td><td>MT1W_US</td><td>Untyped Surface Write message</td></tr> <tr><td>0Ah</td><td>MT1W_MB</td><td>Media Block Write message</td></tr> <tr><td>0Bh</td><td>MT1A_TC</td><td>Typed Atomic Counter Operation message</td></tr> <tr><td>0Ch</td><td>MT1A_TCH</td><td>Typed Atomic Half Counter Operation message</td></tr> <tr><td>0Dh</td><td>MT1W_TS</td><td>Typed Surface Write message</td></tr> <tr><td>0Eh</td><td>Reserved</td><td>Ignored</td></tr> <tr><td>10h</td><td>MT1R_A64_SB</td><td>A64 Scattered Read message</td></tr> <tr><td>11h</td><td>MT1R_A64_US</td><td>A64 Untyped Surface Read message</td></tr> <tr><td>12h</td><td>MT1A_A64_UI</td><td>A64 Untyped Atomic Integer Operation message</td></tr> <tr><td>13h</td><td>MT1A_A64_UHI</td><td>A64 Untyped Atomic Half Integer Operation message</td></tr> <tr><td>14h</td><td>MT1R_A64_B</td><td>A64 Block Read message</td></tr> <tr><td>15h</td><td>MT1W_A64_B</td><td>A64 Block Write message</td></tr> <tr><td>18h</td><td>Reserved</td><td>Ignored</td></tr> <tr><td>19h</td><td>MT1W_A64_US</td><td>A64 Untyped Surface Write message</td></tr> <tr><td>1Ah</td><td>MT1W_A64_SB</td><td>A64 Scattered Write message</td></tr> <tr><td>1Bh</td><td>MT1A_UF</td><td>Untyped Atomic Float Operation message</td></tr> </tbody> </table>	Value	Name	Description	01h	MT1R_US	Untyped Surface Read message	02h	MT1A_UI	Untyped Atomic Integer Operation message	03h	MT1A_UHI	Untyped Atomic Half Integer Operation 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_TAH	Typed Atomic Half Integer Operation 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_TCH	Typed Atomic Half Counter Operation 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_UHI	A64 Untyped Atomic Half Integer Operation 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	1Ah	MT1W_A64_SB	A64 Scattered Write message	1Bh	MT1A_UF	Untyped Atomic Float Operation message
Value	Name	Description																																																																											
01h	MT1R_US	Untyped Surface Read message																																																																											
02h	MT1A_UI	Untyped Atomic Integer Operation message																																																																											
03h	MT1A_UHI	Untyped Atomic Half Integer Operation 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_TAH	Typed Atomic Half Integer Operation 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_TCH	Typed Atomic Half Counter Operation 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_UHI	A64 Untyped Atomic Half Integer Operation 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																																																																											
1Ah	MT1W_A64_SB	A64 Scattered Write message																																																																											
1Bh	MT1A_UF	Untyped Atomic Float Operation message																																																																											



**MT\_DP1 - Data Port 1 Message Types**

		1Ch	MT1A_UHF	Untyped Atomic Half Float Operation message
		1Dh	MT1A_A64_UF	A64 Untyped Atomic Float Operation message
		1Eh	MT1A_A64_UHF	A64 Untyped Atomic Half Float Operation message
		Others	Reserved	Ignored



## Data Port Bindless Surface Extended Message Descriptor

DP_EXTDESC_BTI252 - Data Port Bindless Surface Extended Message Descriptor					
Size (in bits):		32			
Default Value:		0x00000000			
DWord	Bit	Description			
0	31:12	<b>Bindless Surface Offset</b>			
		<table border="1"> <tr> <td>Format:</td> <td>SurfaceStateOffset[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:	SurfaceStateOffset[25:6]	
	Format:	SurfaceStateOffset[25:6]			
	11	<b>Reserved</b>			
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
Format:	MBZ				
10:0	<b>Reserved</b>				
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				

## Data Size Message Descriptor Control Field

MDC_DS - Data Size Message Descriptor Control Field																	
Size (in bits):		2															
Default Value:		0x00000000															
DWord	Bit	Description															
0	1:0	<b>Data Size</b> Specifies the number of Bytes to be read or written <table border="1" data-bbox="418 604 1466 829"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>B</td> <td>1 Byte</td> </tr> <tr> <td>01h</td> <td>W</td> <td>2 Bytes</td> </tr> <tr> <td>02h</td> <td>DW</td> <td>4 Bytes</td> </tr> <tr> <td>03h</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	00h	B	1 Byte	01h	W	2 Bytes	02h	DW	4 Bytes	03h	Reserved	Reserved
Value	Name	Description															
00h	B	1 Byte															
01h	W	2 Bytes															
02h	DW	4 Bytes															
03h	Reserved	Reserved															



## Depth Clear Value Format

Depth Clear Value Format		
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0 This field defines the clear value that will be applied to the depth buffer if the Depth Buffer Clear field is enabled. It is valid only if Depth Buffer Clear Value Valid is set. <b>Programming Notes:</b> The clear value must be between the min and max depth values (inclusive) defined in the CC_VIEWPORT. If the depth buffer format is D32_FLOAT, then values must be limited to the range of +0.0f and 1.0f inclusive; values outside this range are reserved	31:0	<b>Depth Clear Value</b> Format: IEEE_FLOAT

## Depth Clear Value Format

STRUCTURE_TEMPLATE - Depth Clear Value Format		
Size (in bits):		32
Default Value:		0x00000000
DWord	Bit	Description
0	31:0	<b>Address1</b>
		Format: IEEE_FLOAT32
		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)

## DirectOperand

DirectOperand											
Size (in bits):	14										
Default Value:	0x00000000										
DWord	Bit	Description									
0	13:6	<b>RegNum</b> Format: <span style="float: right;">U8</span>									
		<b>Description</b> This field provide 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 destination or Source 0. Any Source 1 or Source 2 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. This field applies to both source and destination operands.									
		<b>SubRegNum</b> This field provide 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 Word-aligned; SubRegNum provides bits 4:1 of the address and bits 0 are zero.									
0	0	<b>RegFile</b> This field indicate whether Architecture register file or General register file are selected.									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>ARF</td> <td>Architecture Register File.</td> </tr> <tr> <td>1</td> <td>GRF</td> <td>General Register File. Allowed for any Source or Destination.</td> </tr> </tbody> </table>	Value	Name	Description	0	ARF	Architecture Register File.	1	GRF	General Register File. Allowed for any Source or Destination.
		Value	Name	Description							
0	ARF	Architecture Register File.									
1	GRF	General Register File. Allowed for any Source or Destination.									

## Display Engine Render Response Message Definition

DE_RRMD - Display Engine Render Response Message Definition			
Size (in bits):	96		
Default Value:	0x00000000, 0x00000000, 0x00000000		
The Display Engine Render Response Registers use bit definitions from this table.			
Programming Notes			
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.			
The STEREO3D_EVENT_MASK selects between left eye and right eye reporting of vertical blank and scanline events in stereo 3D modes.			
DWord	Bit	Description	
0	31	<b>Spare 31</b>	
	30	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	29	<b>Reserved</b>	
	28	<b>Spare 28</b>	
	27	<b>Spare 27</b>	
	26	<b>VDenc Session3 Start</b> This event is reported when the Display pipe for this session reaches Tail Initial Update Delay. This event is not masked by DE_GUCRMR since it is sent directly by transcoder WD.	
	25	<b>VDenc Session2 Start</b> This event is reported when the Display pipe for this session reaches Tail Initial Update Delay. This event is not masked by DE_GUCRMR since it is sent directly by transcoder WD.	
24	<b>VDenc Session1 Start</b> This event is reported when the Display pipe for this session reaches Tail Initial Update Delay. This event is not masked by DE_GUCRMR since it is sent directly by transcoder WD.		
23	<b>VDenc Session0 Start</b> This event is reported when the Display pipe for this session reaches Tail Initial Update Delay. This event is not masked by DE_GUCRMR since it is sent directly by transcoder WD.		

## DE\_RRMD - Display Engine Render Response Message Definition

22	<b>Pipe_C_DPST_Histogram_Event</b> This event is reported on the start of the DPST Histogram event for pipe C. This event should only be unmasked for GuC.
21	<b>Pipe_C_Start_of_Vertical_Blank_Event</b> This event is reported on the start of the vertical blank of the transcoder attached to Pipe C.
20	<b>Plane_6_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 6.
19	<b>Plane_12_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 12.
18	<b>Plane_11_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 11.
17	<b>Plane_10_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 10.
16	<b>Plane_9_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 9.
15	<b>Plane_3_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 3.
14	<b>Pipe_C_Scanline_Event</b> This event is reported on the start of the selected scan line for the transcoder attached to Pipe C.
13	<b>Reserved</b>
12	<b>Spare 12</b> Unused
11	<b>Pipe_B_Start_of_Vertical_Blank_Event</b> This event is reported on the start of the vertical blank of the transcoder attached to Pipe B.
10	<b>Plane_5_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 5.
9	<b>Plane_2_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 2.
8	<b>Pipe_B_Scanline_Event</b> This event is reported on the start of the selected scan line for the transcoder attached to Pipe B.
7	<b>Plane_8_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 8.
6	<b>Plane_7_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 7.
5	<b>Reserved</b>
4	<b>Spare 4</b> Unused
3	<b>Pipe_A_Start_of_Vertical_Blank_Event</b> This event is reported on the start of the vertical blank of the transcoder attached to Pipe A.
2	<b>Plane_4_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 4.



<b>DE_RRMD - Display Engine Render Response Message Definition</b>		
	1	<b>Plane_1_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 1.
	0	<b>Pipe_A_Scanline_Event</b> This event is reported on the start of the selected scan line for the transcoder attached to Pipe A.
1	31	<b>Spare 31</b> Unused.
	30	<b>Spare 30</b> Unused.
	29	<b>Spare 29</b> Unused.
	28	<b>Spare 28</b> Unused
	27	<b>Spare 27</b> Unused
	26	<b>Spare 26</b> Unused
	25	<b>Spare 25</b> Unused
	24	<b>Spare 24</b> Unused
	23	<b>Spare 23</b> Unused
	22	<b>Spare 22</b> Unused
	21	<b>Spare 21</b> Unused
	20	<b>Spare 20</b> Unused
	19	<b>Spare 19</b> Unused
	18	<b>Spare 18</b> Unused
17	<b>Spare 17</b> Unused	
16	<b>Spare 16</b> Unused	
15	<b>Spare 15</b> Unused	
14	<b>Spare 14</b> Unused	

<b>DE_RRMD - Display Engine Render Response Message Definition</b>		
	13	<b>Spare 13</b> Unused
	12	<b>Spare 12</b> Unused
	11	<b>Spare 11</b> Unused
	10	<b>Spare 10</b> Unused
	9	<b>Spare 9</b> Unused
	8	<b>Spare 8</b> Unused
	7	<b>Spare 7</b> Unused
	6	<b>Spare 6</b> Unused
	5	<b>Spare 5</b> Unused
	4	<b>Spare 4</b> Unused
	3	<b>Spare 3</b> Unused
	2	<b>Reserved</b>
	1	<b>Pipe_D_Scanline_Event</b> This event is reported on the start of the selected scan line for the transcoder attached to Pipe D. Some SKUs may not have Pipe D.
0	<b>Pipe_D_Start_of_Vertical_Blank_Event</b> This event is reported on the start of the vertical blank of the transcoder attached to Pipe D. Some SKUs may not have Pipe D.	
2	31	<b>Spare 31</b> Unused.
	30	<b>Spare 30</b> Unused.
	29	<b>Spare 29</b> Unused.
	28	<b>Spare 28</b> Unused
	27	<b>Spare 27</b> Unused
	26	<b>Spare 26</b> Unused

<b>DE_RRMD - Display Engine Render Response Message Definition</b>	
25	<b>Spare 25</b> Unused
24	<b>Spare 24</b> Unused
23	<b>Spare 23</b> Unused
22	<b>Spare 22</b> Unused
21	<b>Spare 21</b> Unused
20	<b>Spare 20</b> Unused
19	<b>Plane_32_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 32.
18	<b>Plane_31_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 31.
17	<b>Plane_30_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 30.
16	<b>Plane_29_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 29.
15	<b>Plane_28_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 28.
14	<b>Plane_27_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 27.
13	<b>Plane_26_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 26.
12	<b>Plane_25_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 25.
11	<b>Plane_24_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 24.
10	<b>Plane_23_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 23.
9	<b>Plane_22_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 22.
8	<b>Plane_21_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 21.
7	<b>Plane_20_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 20.
6	<b>Plane_19_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 19.

## DE\_RRMD - Display Engine Render Response Message Definition

5	<b>Plane_18_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 18.
4	<b>Plane_17_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 17.
3	<b>Plane_16_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 16.
2	<b>Plane_15_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 15.
1	<b>Plane_14_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 14.
0	<b>Plane_13_Flip_Done_Event</b> This event is reported on the completion of a flip for Plane 13.

## DUALSUBSLICE\_HASH\_TABLE\_8x8

DUALSUBSLICE_HASH_TABLE_8x8				
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
8x8 [Y][X] dualsubslice hashing table. Each entry is a single bit that indicates which dualSubSlice(DSS) the indicated xy location maps to. A value of 0 indicates the larger DSS, or DSS=0 if both DSS have are balanced(have same number of enabled lsubslices)				
DWord	Bit	Description		
0	31:24	<b>SubSlice Hashing Table Entries[3]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=3 and x=7..0	Format:	U8
	Format:	U8		
	23:16	<b>SubSlice Hashing Table Entries[2]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=2 and x=7..0	Format:	U8
	Format:	U8		
15:8	<b>SubSlice Hashing Table Entries[1]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=1 and x=7..0	Format:	U8	
Format:	U8			
7:0	<b>SubSlice Hashing Table Entries[0]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=0 and x=7..0	Format:	U8	
Format:	U8			
1	31:24	<b>SubSlice Hashing Table Entries[7]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=7 and x=7..0	Format:	U8
	Format:	U8		
	23:16	<b>SubSlice Hashing Table Entries[6]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=6 and x=7..0	Format:	U8
	Format:	U8		
15:8	<b>SubSlice Hashing Table Entries[5]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=5 and x=7..0	Format:	U8	
Format:	U8			
7:0	<b>SubSlice Hashing Table Entries[4]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=4 and x=7..0	Format:	U8	
Format:	U8			



## DUALSUBSLICE\_HASH\_TABLE\_16x8

<b>DUALSUBSLICE_HASH_TABLE_16x8</b>				
Size (in bits):	128			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000			
<p>16x8 [Y][X] dualsubslice hashing table. Each entry is a single bit that indicates which dualSubSlice(DSS) the indicated xy location maps to. A value of 0 indicates the larger DSS, or DSS=0 if both DSS have are balanced(have same number of enabled lsubslices)</p>				
DWord	Bit	Description		
0	31:16	<b>SubSlice Hashing Table Entries y[1]x[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=1 and x=15..0	Format:	U16
	Format:	U16		
15:0	<b>SubSlice Hashing Table Entries y[0]x[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=0 and x=15..0	Format:	U16	
Format:	U16			
1	31:16	<b>SubSlice Hashing Table Entries y[3]x[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=3 and x=15..0	Format:	U16
	Format:	U16		
15:0	<b>SubSlice Hashing Table Entries y[2]x[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=2 and x=15..0	Format:	U16	
Format:	U16			
2	31:16	<b>SubSlice Hashing Table Entries y[5]x[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=5 and x=15..0	Format:	U16
	Format:	U16		
15:0	<b>SubSlice Hashing Table Entries y[4]x[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=4 and x=15..0	Format:	U16	
Format:	U16			
3	31:16	<b>SubSlice Hashing Table Entries y[7]x[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=7 and x=15..0	Format:	U16
	Format:	U16		
15:0	<b>SubSlice Hashing Table Entries y[6]x[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Indicates the dualsubslice_id for the pixel block that has y=6 and x=15..0	Format:	U16	
Format:	U16			

## Dword Data Payload Register

<b>MDCR_DW - Dword Data Payload Register</b>				
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0	31:0	<b>Dword0</b> <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	<b>Dword1</b> <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	<b>Dword2</b> <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	<b>Dword3</b> <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	<b>Dword4</b> <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	<b>Dword5</b> <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	<b>Dword6</b> <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	<b>Dword7</b> <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 SIMD8 Atomic Operation CMPWR Message Data Payload

<b>MDP_AOP8_DW2 - Dword SIMD8 Atomic Operation CMPWR Message Data Payload</b>		
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	<b>Src0</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_DW</b></td> </tr> </table> <p>Specifies the Slot [7:0] Source 0 data</p>
Format:	<b>MDCR_DW</b>	
1.0-1.7	255:0	<b>Src1</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_DW</b></td> </tr> </table> <p>Specifies the Slot [7:0] Source 1 data</p>
Format:	<b>MDCR_DW</b>	



## Dword SIMD8 Data Payload

MDP_DW_SIMD8 - Dword SIMD8 Data Payload		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Data[7:0]</b> Format: <b>MDCR_DW</b> Specifies the Slot [7:0] data

## Dword SIMD16 Atomic Operation CMPWR Message Data Payload

<b>MDP_AOP16_DW2 - Dword SIMD16 Atomic Operation CMPWR Message Data Payload</b>				
Size (in bits):	1024			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<b>Src0[7:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;"><b>MDCR_DW</b></td> </tr> </table> Specifies the Source 0 data for Slot [7:0]	Format:	<b>MDCR_DW</b>
Format:	<b>MDCR_DW</b>			
1.0-1.7	255:0	<b>Src0[15:8]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;"><b>MDCR_DW</b></td> </tr> </table> Specifies the Source 0 data for Slot [15:8]	Format:	<b>MDCR_DW</b>
Format:	<b>MDCR_DW</b>			
2.0-2.7	255:0	<b>Src1[7:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;"><b>MDCR_DW</b></td> </tr> </table> Specifies the Source 1 data for Slot [7:0]	Format:	<b>MDCR_DW</b>
Format:	<b>MDCR_DW</b>			
3.0-3.7	255:0	<b>Src1[15:8]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;"><b>MDCR_DW</b></td> </tr> </table> Specifies the Source 1 data for Slot [15:8]	Format:	<b>MDCR_DW</b>
Format:	<b>MDCR_DW</b>			

## Dword SIMD16 Data Payload

<b>MDP_DW_SIMD16 - Dword SIMD16 Data Payload</b>				
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	<b>Data[7:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_DW</b></td> </tr> </table> Specifies the Slot [7:0] data	Format:	<b>MDCR_DW</b>
Format:	<b>MDCR_DW</b>			
1.0-1.7	255:0	<b>Data[15:8]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_DW</b></td> </tr> </table> Specifies the Slot [15:8] data	Format:	<b>MDCR_DW</b>
Format:	<b>MDCR_DW</b>			

## Encoder Statistics Format

Encoder Statistics Format						
Source:	VideoEnhancementCS					
Size (in bits):	128					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000					
<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	<b>Tearing_Count 1 (FMD Variance[8])</b>				
		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		<b>Tearing_Count 2</b>				
		Format: U8				
		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)				
		If the frame is bottom first then this is (FMD Variance[10]) = Number of pixels that have (diff_cBpT > diff_pTpT + diff_cBcB)				
15:8		<b>Motion_Count (FMD Variance[7])</b>				
		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		<b>Reserved</b>				
		Access: RO				
		Format: MBZ				

Encoder Statistics Format														
1	31:28	<p><b>sSTAD</b></p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">U4</td> </tr> <tr> <td colspan="3">Shift for the Sum in time of absolute differences for 16x4.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>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><b>sSHCM</b></p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">U4</td> </tr> <tr> <td colspan="3">Shift for the Sum horizontally of absolute differences.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>0</td> <td></td> <td>DN is Disabled</td> </tr> </table>	Format:	U4		Shift for the Sum horizontally of absolute differences.			Value	Name	Description	0		DN is Disabled
	Format:	U4												
Shift for the Sum horizontally of absolute differences.														
Value	Name	Description												
0		DN is Disabled												
23:20	<p><b>sSVCM</b></p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">U4</td> </tr> <tr> <td colspan="3">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><b>sDiff_cTpT</b></p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">U4</td> </tr> <tr> <td colspan="3">Shift for the sum of differences in top fields of current and previous frame.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>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><b>sDiff_cBpB</b></p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">U4</td> </tr> <tr> <td colspan="3">Shift for the sum of differences in bottom field of current and previous frame.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>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><b>sDiff_cTcB</b></p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">U4</td> </tr> <tr> <td colspan="3">Shift for the sum of differences between top and bottom field in current frame.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>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><b>sDiff_cTpB</b></p> <table border="1"> <tr> <td>Format:</td> <td colspan="2">U4</td> </tr> <tr> <td colspan="3">Shift for the sum of differences between current top and previous bottom.</td> </tr> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> <tr> <td>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												

<b>Encoder Statistics Format</b>								
2	3:0	<b>sDiff_cBpT</b> 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	<b>mDiff_cBpB (FMD Variance[1])</b> 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	<b>mDiff_cTcB (FMD Variance[2])</b> 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	<b>mDiff_cTpB (FMD Variance[3])</b> 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	<b>mDiff_cBpT (FMD Variance[4])</b> 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	<b>mSTAD</b> 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	<b>mSHCM</b> Format: U8 Mantissa of Sum horizontally 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						

<b>Encoder Statistics Format</b>						
	15:8	<b>mSVCM</b>				
		Format: <span style="float: right;">U8</span>				
		Mantissa of Sum vertically of absolute differences.				
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>DN is Disabled</td> </tr> </tbody> </table>	Value	Name	Description	0
Value	Name	Description				
0		DN is Disabled				
	7:0	<b>mDiff_cTpT (FMD Variance[0])</b>				
		Format: <span style="float: right;">U8</span>				
		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="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>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	127:96	<b>Src0.ImmValue[31:0]</b> Exists If: $[[Src0.IsImm] == true]$
	95:92	<b>CondCtrl</b> Exists If: $[[Src0.IsImm] == false] \text{ OR } ([[Src0.DataType] != :q] \text{ AND } [[Src0.DataType] != :uq] \text{ AND } [[Src0.DataType] != :df])$ Format: <b>FlagModifier</b>
	95:64	<b>Src0.ImmValue[63:32]</b> Exists If: $[[Src0.IsImm] == true] \text{ AND } ([[Src0.DataType] == :q] \text{ OR } [[Src0.DataType] == :uq] \text{ OR } [[Src0.DataType] == :df])$
	87:84	<b>Src0.VertStride</b> Exists If: $[[Src0.IsImm] == false] \text{ OR } ([[Src0.DataType] != :q] \text{ AND } [[Src0.DataType] != :uq] \text{ AND } [[Src0.DataType] != :df])$ Format: <b>VertStride</b>
	83:81	<b>Src0.Width</b> Exists If: $[[Src0.IsImm] == false] \text{ OR } ([[Src0.DataType] != :q] \text{ AND } [[Src0.DataType] != :uq] \text{ AND } [[Src0.DataType] != :df])$ Format: <b>Width</b>
	80	<b>Src0.AddrMode</b> Exists If: $[[Src0.IsImm] == false] \text{ OR } ([[Src0.DataType] != :q] \text{ AND } [[Src0.DataType] != :uq] \text{ AND } [[Src0.DataType] != :df])$ Format: <b>AddrMode</b>
	79:66	<b>Src0.Operand</b> Exists If: $((([Src0.IsImm] == false) \text{ OR } ([[Src0.DataType] != :q] \text{ AND } [[Src0.DataType] != :uq] \text{ AND } [[Src0.DataType] != :df]))) \text{ AND } [[Src0.AddrMode] == \text{Direct}]$ Format: <b>DirectOperand</b>
	79:66	<b>Src0.Operand</b> Exists If: $((([Src0.IsImm] == false) \text{ OR } ([[Src0.DataType] != :q] \text{ AND } [[Src0.DataType] != :uq] \text{ AND } [[Src0.DataType] != :df]))) \text{ AND } [[Src0.AddrMode] == \text{Indirect}]$ Format: <b>IndirectOperand</b>



## EU\_INSTRUCTION\_BASIC\_ONE\_SRC

65:64	<b>Src0.HorzStride</b>	
	Exists If:	(([Src0.IsImm]==false) OR ((([Src0.DataType]!=:q) AND ([Src0.DataType]!=:uq) AND ([Src0.DataType]!=:df)))
	Format:	<b>HorzStride</b>
63:50	<b>Dst.Operand</b>	
	Exists If:	([Dst.AddrMode]==Indirect)
	Format:	<b>IndirectOperand</b>
63:50	<b>Dst.Operand</b>	
	Exists If:	([Dst.AddrMode]==Direct)
	Format:	<b>DirectOperand</b>
49:48	<b>Dst.HorzStride</b>	
	Format:	<b>HorzStride</b>
47	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
46	<b>Src0.IsImm</b>	
	This field indicate that Source 0 operand is carrying an immediate value.	
	<b>Value</b>	<b>Name</b>
	0	false <b>[Default]</b>
	1	true
45:44	<b>Src0.Mod</b>	
	Format:	<b>SrcMod</b>
43:40	<b>Src0.DataType</b>	
	Exists If:	([Src0.IsImm]==false)
	Format:	<b>RegDataType</b>
43:40	<b>Src0.DataType</b>	
	Exists If:	([Src0.IsImm]==true)
	Format:	<b>ImmDataType</b>
39:36	<b>Dst.DataType</b>	
	Format:	<b>RegDataType</b>
35	<b>Dst.AddrMode</b>	
	Format:	<b>AddrMode</b>
34	<b>Saturate</b>	
	Format:	<b>Saturate</b>
33	<b>AccWrCtrl</b>	
	Format:	<b>AccWrCtrl</b>

<b>EU_INSTRUCTION_BASIC_ONE_SRC</b>											
32	<b>AtomicCtrl</b> Format: <span style="float: right;"><b>AtomicCtrl</b></span>										
31	<b>MaskCtrl</b> Mask Control (formerly Write Enable Control). This field determines if the per channel write enables are used to generate the final write enable. This field should be normally "0". <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Normal <b>[Default]</b></td> <td>Normal. Per channel write enable used for final write enable generation.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>NoMask</td> <td>NoMask.Skips the check for PclP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.</td> </tr> </tbody> </table>		Value	Name	Description	0	Normal <b>[Default]</b>	Normal. Per channel write enable used for final write enable generation.	1	NoMask	NoMask.Skips the check for PclP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.
Value	Name	Description									
0	Normal <b>[Default]</b>	Normal. Per channel write enable used for final write enable generation.									
1	NoMask	NoMask.Skips the check for PclP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.									
30	<b>Reserved</b>										
29	<b>CmptCtrl</b> Format: <span style="float: right;">MBZ</span> 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, the variations supported by those lookup tables and the compact format. See EU Compact Instruction Format for more information. <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>NoCompaction <b>[Default]</b></td> <td>No compaction. 128-bit native instruction supporting all instruction options.</td> </tr> <tr> <td style="text-align: center;">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 <b>[Default]</b>	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 <b>[Default]</b>	No compaction. 128-bit native instruction supporting all instruction options.									
1	Compacted	Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.									
28	<b>PredInv</b> 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 <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>Positive <b>[Default]</b></td> <td>Positive polarity of predication. Use the predication mask produced by PredCtrl.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Negative</td> <td>Negative polarity of predication. If PredCtrl is nonzero, invert the predication mask.</td> </tr> </tbody> </table>		Value	Name	Description	0	Positive <b>[Default]</b>	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.
Value	Name	Description									
0	Positive <b>[Default]</b>	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.									
27:24	<b>PredCtrl</b> Format: <span style="float: right;"><b>PredCtrl</b></span> 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.										

<b>EU_INSTRUCTION_BASIC_ONE_SRC</b>			
23	<p><b>FlagRegNum[0]</b> This field specifies bit[0] of the register number for a flag register operand.</p>		
22	<p><b>FlagSubRegNum</b> This field specifies the sub-register number for a flag register operand. There are two sub-registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.</p>		
21:19	<p><b>ChanOff</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;"><b>ChanOff</b></td> </tr> </table> <p>This field provides offset information for ARF selection. The can be thought of as a starting channel offset for the execution mask and other ARF registers implicitly accessed.</p>	Format:	<b>ChanOff</b>
Format:	<b>ChanOff</b>		
18:16	<p><b>ExecSize</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;"><b>ExecSize</b></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:	<b>ExecSize</b>
Format:	<b>ExecSize</b>		
15:0	<p><b>Header</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;"><b>Header</b></td> </tr> </table>	Format:	<b>Header</b>
Format:	<b>Header</b>		

## 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	127:114	<b>Src2.Operand</b>	
		Exists If: ([Src2.IsImm]==false) AND ([Header][Opcode]!=madm)	
			Format: <b>DirectOperand</b>
	127:114	<b>Src2.Operand</b>	
		Exists If: ([Src2.IsImm]==false) AND ([Header][Opcode]==madm)	
			Format: <b>MacroOperand</b>
	127:112	<b>Src2.ImmValue[15:0]</b>	
		Exists If: ([Src2.IsImm]==true)	
	113:112	<b>Src2.HorzStride</b>	
		Exists If: ([Src2.IsImm]==false)	
			Format: <b>HorzStride</b>
	111:98	<b>Src1.Operand</b>	
		Exists If: ([Header][Opcode]!=madm)	
			Format: <b>DirectOperand</b>
111:98	<b>Src1.Operand</b>		
	Exists If: ([Header][Opcode]==madm)		
		Format: <b>MacroOperand</b>	
97:96	<b>Src1.HorzStride</b>		
	Format: <b>HorzStride</b>		
95:92	<b>CondCtrl</b>		
	Format: <b>FlagModifier</b>		
91	<b>Src1.VertStride[1]</b>		
	Format: <b>TernaryVertStride[1:1]</b>		
90:88	<b>Src1.DataType</b>		
	Format: <b>TernaryDataType</b>		
87:86	<b>Src1.Mod</b>		
	Format: <b>SrcMod</b>		
85:84	<b>Src2.Mod</b>		
	Format: <b>SrcMod</b>		

## EU\_INSTRUCTION\_BASIC\_THREE\_SRC

83	<b>Src1.VertStride[0]</b>	
	Format:	<b>TernaryVertStride[0:0]</b>
82:80	<b>Src2.DataType</b>	
	Format:	<b>TernaryDataType</b>
79:66	<b>Src0.Operand</b>	
	Exists If:	(([Src0.IsImm]==false) AND ([Header][Opcode]!=madm)
	Format:	<b>DirectOperand</b>
79:66	<b>Src0.Operand</b>	
	Exists If:	(([Src0.IsImm]==false) AND ([Header][Opcode]==madm)
	Format:	<b>MacroOperand</b>
79:64	<b>Src0.ImmValue[15:0]</b>	
	Exists If:	(([Src0.IsImm]==true)
65:64	<b>Src0.HorzStride</b>	
	Exists If:	(([Src0.IsImm]==false)
	Format:	<b>HorzStride</b>
63:50	<b>Dst.Operand</b>	
	Exists If:	(([Header][Opcode]!=madm)
	Format:	<b>DirectOperand</b>
	<b>Programming Notes</b>	
	The Dst.Operand must be 64 bit aligned. i.e. Dst.Operand.SubRegNum[2:0] must be zero,	
63:50	<b>Dst.Operand</b>	
	Exists If:	(([Header][Opcode]==madm)
	Format:	<b>MacroOperand</b>
49	<b>Reserved</b>	
	Format:	MBZ
48	<b>Dst.HorzStride</b>	
	This field provides the distance in unit of data elements between two adjacent data elements within a row (horizontal) in the register region for the operand.	
	<b>Value</b>	<b>Name</b>
	0	1 element
	1	2 element
47	<b>Src2.IsImm</b>	
	This field indicate that Source 2 operand is carrying an immediate value.	
	<b>Value</b>	<b>Name</b>
	0	false
	1	true

## EU\_INSTRUCTION\_BASIC\_THREE\_SRC

46	<b>Src0.IsImm</b>	This field indicate that Source 0 operand is carrying an immediate value.	
		<b>Value</b>	<b>Name</b>
		0	false
		1	true
45:44	<b>Src0.Mod</b>	Format:	<b>SrcMod</b>
43	<b>Src0.VertStride[1]</b>	Format:	<b>TernaryVertStride[1:1]</b>
42:40	<b>Src0.DataType</b>	Format:	<b>TernaryDataType</b>
39	<b>ExecDataType</b>	This field indicate the datatype mode of ternary instruction. Integer or Float.	
		<b>Value</b>	<b>Name</b>
		0	Integer
		1	Float
38:36	<b>Dst.DataType</b>	Format:	<b>TernaryDataType</b>
35	<b>Src0.VertStride[0]</b>	Format:	<b>TernaryVertStride[0:0]</b>
34	<b>Saturate</b>	Format:	<b>Saturate</b>
33	<b>AccWrCtrl</b>	Format:	<b>AccWrCtrl</b>
32	<b>AtomicCtrl</b>	Format:	<b>AtomicCtrl</b>
31	<b>MaskCtrl</b>	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".	
		<b>Value</b>	<b>Name</b>
		0	Normal <b>[Default]</b>
		1	NoMask
			<b>Description</b>
			Normal. Per channel write enable used for final write enable generation.
			NoMask.Skips the check for PciP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.
30	<b>Reserved</b>		
29	<b>CmptCtrl</b>	Format:	MBZ

## EU\_INSTRUCTION\_BASIC\_THREE\_SRC

	<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, 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 <b>[Default]</b></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 <b>[Default]</b>	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 <b>[Default]</b>	No compaction. 128-bit native instruction supporting all instruction options.								
1	Compacted	Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.								
28	<p><b>PredInv</b> 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 <b>[Default]</b></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>	Value	Name	Description	0	Positive <b>[Default]</b>	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.
Value	Name	Description								
0	Positive <b>[Default]</b>	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.								
27:24	<p><b>PredCtrl</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>PredCtrl</b></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.</p>	Format:	<b>PredCtrl</b>							
Format:	<b>PredCtrl</b>									
23	<p><b>FlagRegNum[0]</b> This field specifies bit[0] of the register number for a flag register operand.</p>									
22	<p><b>FlagSubRegNum</b> This field specifies the sub-register number for a flag register operand. There are two sub-registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.</p>									
21:19	<p><b>ChanOff</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>ChanOff</b></td> </tr> </table> <p>This field provides offset information for ARF selection. The can be thought of as a starting channel offset for the execution mask and other ARF registers implicitly accessed.</p>	Format:	<b>ChanOff</b>							
Format:	<b>ChanOff</b>									

<b>EU_INSTRUCTION_BASIC_THREE_SRC</b>		
	18:16	<b>ExecSize</b> Format: <span style="float: right;"><b>ExecSize</b></span> 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.
	15:0	<b>Header</b> Format: <span style="float: right;"><b>Header</b></span>



## 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	127:126	<b>Reserved</b>
		Exists If: ([Src1.IsImm]==false)
		Format: MBZ
	127:96	<b>Src1.ImmValue[31:0]</b>
		Exists If: ([Src1.IsImm]==true)
	125:122	<b>Reserved</b>
		Exists If: ([Src1.IsImm]==false)
		Format: MBZ
	121:120	<b>Src1.Mod</b>
		Exists If: ([Src1.IsImm]==false)
		Format: <b>SrcMod</b>
	119:116	<b>Src1.VertStride</b>
		Exists If: ([Src1.IsImm]==false)
		Format: <b>VertStride</b>
	115:113	<b>Src1.Width</b>
		Exists If: ([Src1.IsImm]==false)
Format: <b>Width</b>		
112	<b>Src1.AddrMode</b>	
	Exists If: ([Src1.IsImm]==false)	
	Format: <b>AddrMode</b>	
111:98	<b>Src1.Operand</b>	
	Exists If: ([Src1.IsImm]==false) AND ([Src1.AddrMode]==Indirect)	
	Format: <b>IndirectOperand</b>	
111:98	<b>Src1.Operand</b>	
	Exists If: ([Src1.IsImm]==false) AND ([Src1.AddrMode]==Direct)	
	Format: <b>DirectOperand</b>	
97:96	<b>Src1.HorzStride</b>	
	Exists If: ([Src1.IsImm]==false)	
	Format: <b>HorzStride</b>	

## EU\_INSTRUCTION\_BASIC\_TWO\_SRC

95:92	<b>CondCtrl</b>	Format:	<b>FlagModifier</b>
91:88	<b>Src1.DataType</b>	Exists If:	((Src1.IsImm)==true)
		Format:	<b>ImmDataType</b>
91:88	<b>Src1.DataType</b>	Exists If:	((Src1.IsImm)==false)
		Format:	<b>RegDataType</b>
87:84	<b>Src0.VertStride</b>	Format:	<b>VertStride</b>
83:81	<b>Src0.Width</b>	Format:	<b>Width</b>
80	<b>Src0.AddrMode</b>	Format:	<b>AddrMode</b>
79:66	<b>Src0.Operand</b>	Exists If:	((Src0.AddrMode)==Direct)
		Format:	<b>DirectOperand</b>
79:66	<b>Src0.Operand</b>	Exists If:	((Src0.AddrMode)==Indirect)
		Format:	<b>IndirectOperand</b>
65:64	<b>Src0.HorzStride</b>	Format:	<b>HorzStride</b>
63:50	<b>Dst.Operand</b>	Exists If:	((Dst.AddrMode)==Direct)
		Format:	<b>DirectOperand</b>
63:50	<b>Dst.Operand</b>	Exists If:	((Dst.AddrMode)==Indirect)
		Format:	<b>IndirectOperand</b>
49:48	<b>Dst.HorzStride</b>	Format:	<b>HorzStride</b>
47	<b>Src1.IsImm</b>	This field indicate that Source 1 operand is carrying an immediate value.	
		<b>Value</b>	<b>Name</b>
		0	false <b>[Default]</b>
		1	true

## EU\_INSTRUCTION\_BASIC\_TWO\_SRC

46	<b>Src0.IsImm</b> This field indicate that Source 0 operand is carrying an immediate value.	
	<b>Value</b>	<b>Name</b>
	0	false <b>[Default]</b>
	1	true
45:44	<b>Src0.Mod</b> Format: <span style="float: right;"><b>SrcMod</b></span>	
43:40	<b>Src0.DataType</b> Exists If: <span style="float: right;">([Src0.IsImm]==false)</span> Format: <span style="float: right;"><b>RegDataType</b></span>	
43:40	<b>Src0.DataType</b> Exists If: <span style="float: right;">([Src0.IsImm]==true)</span> Format: <span style="float: right;"><b>ImmDataType</b></span>	
39:36	<b>Dst.DataType</b> Format: <span style="float: right;"><b>RegDataType</b></span>	
35	<b>Dst.AddrMode</b> Format: <span style="float: right;"><b>AddrMode</b></span>	
34	<b>Saturate</b> Format: <span style="float: right;"><b>Saturate</b></span>	
33	<b>AccWrCtrl</b> Format: <span style="float: right;"><b>AccWrCtrl</b></span>	
32	<b>AtomicCtrl</b> Format: <span style="float: right;"><b>AtomicCtrl</b></span>	
31	<b>MaskCtrl</b> Mask Control (formerly Write Enable Control). This field determines if the per channel write enables are used to generate the final write enable. This field should be normally "0".	
	<b>Value</b>	<b>Name</b>
	0	Normal <b>[Default]</b>
	1	NoMask
		<b>Description</b>
		Normal. Per channel write enable used for final write enable generation.
		NoMask.Skips the check for PcIP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.
30	<b>Reserved</b>	
29	<b>CmptCtrl</b> Format: <span style="float: right;">MBZ</span> 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, the variations	

## EU\_INSTRUCTION\_BASIC\_TWO\_SRC

	supported by those lookup tables and the compact format. See EU Compact Instruction Format for more information.									
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>NoCompaction <b>[Default]</b></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 <b>[Default]</b>	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 <b>[Default]</b>	No compaction. 128-bit native instruction supporting all instruction options.								
1	Compacted	Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.								
28	<p><b>PredInv</b></p> <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 <b>[Default]</b></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>	Value	Name	Description	0	Positive <b>[Default]</b>	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.
Value	Name	Description								
0	Positive <b>[Default]</b>	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.								
27:24	<p><b>PredCtrl</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>PredCtrl</b></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.</p>	Format:	<b>PredCtrl</b>							
Format:	<b>PredCtrl</b>									
23	<p><b>FlagRegNum[0]</b></p> <p>This field specifies bit[0] of the register number for a flag register operand.</p>									
22	<p><b>FlagSubRegNum</b></p> <p>This field specifies the sub-register number for a flag register operand. There are two sub-registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.</p>									
21:19	<p><b>ChanOff</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>ChanOff</b></td> </tr> </table> <p>This field provides offset information for ARF selection. The can be thought of as a starting channel offset for the execution mask and other ARF registers implicitly accessed.</p>	Format:	<b>ChanOff</b>							
Format:	<b>ChanOff</b>									

<b>EU_INSTRUCTION_BASIC_TWO_SRC</b>			
18:16	<p><b>ExecSize</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>ExecSize</b></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:	<b>ExecSize</b>
Format:	<b>ExecSize</b>		
15:0	<p><b>Header</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>Header</b></td> </tr> </table>	Format:	<b>Header</b>
Format:	<b>Header</b>		

## 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	<b>Reserved</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Src0.IsImm]==false)	Format:	MBZ	
	Exists If:	([Src0.IsImm]==false)					
	Format:	MBZ					
	127:96	<b>JIP</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==true)</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> </table> <p>The byte-aligned jump distance if a jump is taken for the channel</p>	Exists If:	([Src0.IsImm]==true)	Format:	S31	
	Exists If:	([Src0.IsImm]==true)					
	Format:	S31					
	95:80	<b>Reserved</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Src0.IsImm]==false)	Format:	MBZ	
	Exists If:	([Src0.IsImm]==false)					
	Format:	MBZ					
	95:64	<b>Reserved</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==true)</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Src0.IsImm]==true)	Format:	MBZ	
	Exists If:	([Src0.IsImm]==true)					
	Format:	MBZ					
	79:66	<b>Src0.Operand</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td><b>DirectOperand</b></td> </tr> </table>	Exists If:	([Src0.IsImm]==false)	Format:	<b>DirectOperand</b>	
	Exists If:	([Src0.IsImm]==false)					
Format:	<b>DirectOperand</b>						
65:64	<b>Reserved</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Src0.IsImm]==false)	Format:	MBZ		
Exists If:	([Src0.IsImm]==false)						
Format:	MBZ						
63:50	<b>Dst.Operand</b> <table border="1"> <tr> <td>Format:</td> <td><b>DirectOperand</b></td> </tr> </table>	Format:	<b>DirectOperand</b>				
Format:	<b>DirectOperand</b>						
49:47	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
Access:	RO						
Format:	MBZ						
46	<b>Src0.IsImm</b> This field indicate that Source 0 operand is carrying an immediate value. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>false</td> </tr> <tr> <td>1</td> <td>true</td> </tr> </tbody> </table>	Value	Name	0	false	1	true
Value	Name						
0	false						
1	true						

## EU\_INSTRUCTION\_BRANCH\_ONE\_SRC

45:34	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
33	<b>BranchCtrl</b> This field is used by <i>goto</i> , <i>if</i> , and <i>else</i> instructions to control branching. See the <b>goto</b> instruction description for more information about BranchCtrl.	
32	<b>AtomicCtrl</b>	
	Format:	<b>AtomicCtrl</b>
31	<b>MaskCtrl</b> Mask Control (formerly Write Enable Control). This field determines if the per channel write enables are used to generate the final write enable. This field should be normally "0".	
	<b>Value</b>	<b>Name</b>
		<b>Description</b>
	0	Normal <b>[Default]</b>
	1	NoMask
		Normal. Per channel write enable used for final write enable generation.
		NoMask.Skips the check for PclP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.
30	<b>Reserved</b>	
29	<b>CmptCtrl</b>	
	Format:	MBZ
	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, the variations supported by those lookup tables and the compact format. See EU Compact Instruction Format for more information.	
	<b>Value</b>	<b>Name</b>
		<b>Description</b>
	0	NoCompaction <b>[Default]</b>
	1	Compacted
		No compaction. 128-bit native instruction supporting all instruction options.
		Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.
28	<b>PredInv</b> 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	
	<b>Value</b>	<b>Name</b>
		<b>Description</b>
	0	Positive <b>[Default]</b>
	1	Negative
		Positive polarity of predication. Use the predication mask produced by PredCtrl.
		Negative polarity of predication. If PredCtrl is nonzero, invert the predication mask.

<b>EU_INSTRUCTION_BRANCH_ONE_SRC</b>			
27:24	<p><b>PredCtrl</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>PredCtrl</b></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.</p>	Format:	<b>PredCtrl</b>
Format:	<b>PredCtrl</b>		
23	<p><b>FlagRegNum[0]</b></p> <p>This field specifies bit[0] of the register number for a flag register operand.</p>		
22	<p><b>FlagSubRegNum</b></p> <p>This field specifies the sub-register number for a flag register operand. There are two sub-registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.</p>		
21:19	<p><b>ChanOff</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>ChanOff</b></td> </tr> </table> <p>This field provides offset information for ARF selection. The can be thought of as a starting channel offset for the execution mask and other ARF registers implicitly accessed.</p>	Format:	<b>ChanOff</b>
Format:	<b>ChanOff</b>		
18:16	<p><b>ExecSize</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>ExecSize</b></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:	<b>ExecSize</b>
Format:	<b>ExecSize</b>		
15:0	<p><b>Header</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>Header</b></td> </tr> </table>	Format:	<b>Header</b>
Format:	<b>Header</b>		



## 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	<b>Reserved</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Src0.IsImm]==false)	Format:	MBZ
	Exists If:	([Src0.IsImm]==false)				
	Format:	MBZ				
	127:96	<b>JIP</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==true)</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> </table> <p>The byte-aligned jump distance if a jump is taken for the channel.</p>	Exists If:	([Src0.IsImm]==true)	Format:	S31
	Exists If:	([Src0.IsImm]==true)				
	Format:	S31				
	95:80	<b>Reserved</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Src0.IsImm]==false)	Format:	MBZ
	Exists If:	([Src0.IsImm]==false)				
	Format:	MBZ				
	95:64	<b>Reserved</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==true) AND ([Src1.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Src0.IsImm]==true) AND ([Src1.IsImm]==false)	Format:	MBZ
	Exists If:	([Src0.IsImm]==true) AND ([Src1.IsImm]==false)				
	Format:	MBZ				
95:64	<b>UIP</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==true) AND ([Src1.IsImm]==true)</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> </table> <p>The byte aligned jump distance if a jump is taken for the instruction.</p>	Exists If:	([Src0.IsImm]==true) AND ([Src1.IsImm]==true)	Format:	S31	
Exists If:	([Src0.IsImm]==true) AND ([Src1.IsImm]==true)					
Format:	S31					
79:66	<b>Src0.Operand</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td><b>DirectOperand</b></td> </tr> </table>	Exists If:	([Src0.IsImm]==false)	Format:	<b>DirectOperand</b>	
Exists If:	([Src0.IsImm]==false)					
Format:	<b>DirectOperand</b>					
65:64	<b>Reserved</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Src0.IsImm]==false)	Format:	MBZ	
Exists If:	([Src0.IsImm]==false)					
Format:	MBZ					
63:50	<b>Dst.Operand</b> <table border="1"> <tr> <td>Format:</td> <td><b>DirectOperand</b></td> </tr> </table>	Format:	<b>DirectOperand</b>			
Format:	<b>DirectOperand</b>					
49:48	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					

## EU\_INSTRUCTION\_BRANCH\_TWO\_SRC

47	<b>Src1.IsImm</b> This field indicate that Source 1 operand is carrying an immediate value	
	<b>Value</b>	<b>Name</b>
	0	false
	1	true
46	<b>Src0.IsImm</b> This field indicate that Source 0 operand is carrying an immediate value	
	<b>Value</b>	<b>Name</b>
	0	false
	1	true
45:34	<b>Reserved</b> Access: RO Format: MBZ	
33	<b>BranchCtrl</b> This field is used by <i>goto</i> , <i>if</i> , and <i>else</i> instructions to control branching. See the <i>goto</i> instruction description for more information about BranchCtrl.	
32	<b>AtomicCtrl</b> Format: <b>AtomicCtrl</b>	
31	<b>MaskCtrl</b> Mask Control (formerly Write Enable Control). This field determines if the per channel write enables are used to generate the final write enable. This field should be normally "0".	
	<b>Value</b>	<b>Name</b>
	0	Normal <b>[Default]</b>
	1	NoMask
		<b>Description</b>
		Normal. Per channel write enable used for final write enable generation.
		NoMask.Skips the check for PclP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.
30	<b>Reserved</b>	
29	<b>CmptCtrl</b> Format: MBZ 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, the variations supported by those lookup tables and the compact format. See EU Compact Instruction Format for more information.	
	<b>Value</b>	<b>Name</b>
	0	NoCompaction <b>[Default]</b>
	1	Compacted
		<b>Description</b>
		No compaction. 128-bit native instruction supporting all instruction options.
		Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.

## EU\_INSTRUCTION\_BRANCH\_TWO\_SRC

28	<p><b>PredInv</b></p> <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" 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>Positive <b>[Default]</b></td> <td>Positive polarity of predication. Use the predication mask produced by PredCtrl.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Negative</td> <td>Negative polarity of predication. If PredCtrl is nonzero, invert the predication mask.</td> </tr> </tbody> </table>	Value	Name	Description	0	Positive <b>[Default]</b>	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.
Value	Name	Description								
0	Positive <b>[Default]</b>	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.								
27:24	<p><b>PredCtrl</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>PredCtrl</b></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.</p>	Format:	<b>PredCtrl</b>							
Format:	<b>PredCtrl</b>									
23	<p><b>FlagRegNum[0]</b></p> <p>This field specifies bit[0] of the register number for a flag register operand.</p>									
22	<p><b>FlagSubRegNum</b></p> <p>This field specifies the sub-register number for a flag register operand. There are two sub-registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.</p>									
21:19	<p><b>ChanOff</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>ChanOff</b></td> </tr> </table> <p>This field provides offset information for ARF selection. The can be thought of as a starting channel offset for the execution mask and other ARF registers implicitly accessed.</p>	Format:	<b>ChanOff</b>							
Format:	<b>ChanOff</b>									
18:16	<p><b>ExecSize</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>ExecSize</b></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:	<b>ExecSize</b>							
Format:	<b>ExecSize</b>									
15:0	<p><b>Header</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>Header</b></td> </tr> </table>	Format:	<b>Header</b>							
Format:	<b>Header</b>									



## 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	<b>Reserved</b>
		Access: RO
	Format: MBZ	
	6:0	<b>Opcode</b>
Format: <b>EU_OPCODE</b>		

## EU\_INSTRUCTION\_MATH

EU_INSTRUCTION_MATH		
Source:	Eulsa	
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..3	127:126	<b>Reserved</b>
		Exists If: (([Src0.IsImm]==false) AND ([Src1.IsImm]==false))
		Format: MBZ
	127:96	<b>Src0.ImmValue[31:0]</b>
		Exists If: (([Src0.IsImm]==true) AND ([Src1.IsImm]==false))
	127:96	<b>Src1.ImmValue[31:0]</b>
		Exists If: (([Src0.IsImm]==false) AND ([Src1.IsImm]==true))
	125:122	<b>Reserved</b>
		Exists If: (([Src0.IsImm]==false) AND ([Src1.IsImm]==false))
		Format: MBZ
	121:120	<b>Src1.Mod</b>
		Exists If: (([Src0.IsImm]==false) AND ([Src1.IsImm]==false))
		Format: <b>SrcMod</b>
	119:116	<b>Src1.VertStride</b>
Exists If: (([Src0.IsImm]==false) AND ([Src1.IsImm]==false))		
Format: <b>VertStride</b>		
115:113	<b>Src1.Width</b>	
	Exists If: (([Src0.IsImm]==false) AND ([Src1.IsImm]==false))	
	Format: <b>Width</b>	
112	<b>Reserved</b>	
	Exists If: (([Src0.IsImm]==false) AND ([Src1.IsImm]==false))	
	Format: MBZ	
111:98	<b>Src1.Operand</b>	
	Exists If: (([Src0.IsImm]==false) AND ([Src1.IsImm]==false) AND ([FuncCtrl]==INVM))	
	Format: <b>MacroOperand</b>	
111:98	<b>Src1.Operand</b>	
	Exists If: (([Src0.IsImm]==false) AND ([Src1.IsImm]==false) AND ([FuncCtrl]!=INVM))	
	Format: <b>DirectOperand</b>	

## EU\_INSTRUCTION\_MATH

97:96	<b>Src1.HorzStride</b>	
	Exists If:	$([Src0.IsImm] == false) \text{ AND } ([Src1.IsImm] == false)$
	Format:	<b>HorzStride</b>
95:92	<b>FuncCtrl</b>	
	Format:	<b>MathFC</b>
91:88	<b>Src1.DataType</b>	
	Exists If:	$([Src0.IsImm] == false) \text{ AND } ([Src1.IsImm] == true)$
	Format:	<b>ImmDataType</b>
91:88	<b>Src1.DataType</b>	
	Exists If:	$([Src0.IsImm] == false) \text{ AND } ([Src1.IsImm] == false)$
	Format:	<b>RegDataType</b>
91:64	<b>Reserved</b>	
	Exists If:	$([Src0.IsImm] == true)$
	Format:	MBZ
87:84	<b>Src0.VertStride</b>	
	Exists If:	$([Src0.IsImm] == false)$
	Format:	<b>VertStride</b>
83:81	<b>Src0.Width</b>	
	Exists If:	$([Src0.IsImm] == false)$
	Format:	<b>Width</b>
80	<b>Reserved</b>	
	Exists If:	$([Src0.IsImm] == false)$
	Format:	MBZ
79:66	<b>Src0.Operand</b>	
	Exists If:	$([Src0.IsImm] == false) \text{ AND } (([FuncCtrl] == INVMM) \text{ OR } ([FuncCtrl] == RSQTM))$
	Format:	<b>MacroOperand</b>
79:66	<b>Src0.Operand</b>	
	Exists If:	$([Src0.IsImm] == false) \text{ AND } (([FuncCtrl] != INVMM) \text{ AND } ([FuncCtrl] != RSQTM))$
	Format:	<b>DirectOperand</b>
65:64	<b>Src0.HorzStride</b>	
	Exists If:	$([Src0.IsImm] == false)$
	Format:	<b>HorzStride</b>
63:50	<b>Dst.Operand</b>	
	Exists If:	$([FuncCtrl] != INVMM) \text{ AND } ([FuncCtrl] != RSQTM)$
	Format:	<b>DirectOperand</b>

## EU\_INSTRUCTION\_MATH

63:50	<b>Dst.Operand</b>	
	Exists If:	((FuncCtrl)==INVM) OR ((FuncCtrl)==RSQTM)
	Format:	<b>MacroOperand</b>
49:48	<b>Dst.HorzStride</b>	
	Format:	<b>HorzStride</b>
47	<b>Src1.IsImm</b>	
	This field indicate that Source 1 operand is carrying an immediate value	
	<b>Value</b>	<b>Name</b>
	0	false
	1	true
46	<b>Src0.IsImm</b>	
	This field indicate that Source 0 operand is carrying an immediate value.	
	<b>Value</b>	<b>Name</b>
	0	false
	1	true
45:44	<b>Src0.Mod</b>	
	Exists If:	((Src0.IsImm)==false)
	Format:	<b>SrcMod</b>
45:44	<b>Reserved</b>	
	Exists If:	((Src0.IsImm)==true)
	Format:	MBZ
43:40	<b>Src0.DataType</b>	
	Exists If:	((Src0.IsImm)==false)
	Format:	<b>RegDataType</b>
43:40	<b>Src0.DataType</b>	
	Exists If:	((Src0.IsImm)==true)
	Format:	<b>ImmDataType</b>
39:36	<b>Dst.DataType</b>	
	Format:	<b>RegDataType</b>
35	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
34	<b>Saturate</b>	
	Format:	<b>Saturate</b>
33	<b>AccWrCtrl</b>	
	Format:	<b>AccWrCtrl</b>

## EU\_INSTRUCTION\_MATH

32	<b>AtomicCtrl</b>	Format: <span style="float: right;"><b>AtomicCtrl</b></span>	
31	<b>MaskCtrl</b>	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".	
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
	0	Normal <b>[Default]</b>	Normal. Per channel write enable used for final write enable generation.
	1	NoMask	NoMask.Skips the check for PciP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.
30	<b>Reserved</b>		
29	<b>CmptCtrl</b>	Format: <span style="float: right;">MBZ</span>	
		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, the variations supported by those lookup tables and the compact format. See EU Compact Instruction Format for more information.	
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
	0	NoCompaction <b>[Default]</b>	No compaction. 128-bit native instruction supporting all instruction options.
	1	Compacted	Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.
28	<b>PredInv</b>	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	
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
	0	Positive <b>[Default]</b>	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.
27:24	<b>PredCtrl</b>	Format: <span style="float: right;"><b>PredCtrl</b></span>	
		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.	



<b>EU_INSTRUCTION_MATH</b>			
23	<p><b>FlagRegNum[0]</b> This field specifies bit[0] of the register number for a flag register operand.</p>		
22	<p><b>FlagSubRegNum</b> This field specifies the sub-register number for a flag register operand. There are two sub-registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.</p>		
21:19	<p><b>ChanOff</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;"><b>ChanOff</b></td> </tr> </table> <p>This field provides offset information for ARF selection. The can be thought of as a starting channel offset for the execution mask and other ARF registers implicitly accessed.</p>	Format:	<b>ChanOff</b>
Format:	<b>ChanOff</b>		
18:16	<p><b>ExecSize</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;"><b>ExecSize</b></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:	<b>ExecSize</b>
Format:	<b>ExecSize</b>		
15:0	<p><b>Header</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: right;"><b>Header</b></td> </tr> </table>	Format:	<b>Header</b>
Format:	<b>Header</b>		

## 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	<b>Reserved</b> Access: RO Format: MBZ
	30	<b>Reserved</b>
	29:28	<b>Reserved</b> Access: RO Format: MBZ
	27:26	<b>Reserved</b> Format: MBZ
	25:18	<b>Reserved</b> Access: RO Format: MBZ
	17:16	<b>Reserved</b> Format: MBZ
	15:0	<b>Header</b> Format: <b>Header</b>

## EU\_INSTRUCTION\_SEND

EU_INSTRUCTION_SEND			
Source:	Eulsa		
Size (in bits):	128		
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000		
DWord	Bit	Description	
0..3	127:124	<b>ExDesc[31:28]</b>	
		Exists If:	([ExDesc.IsReg]==false)
		Format:	<b>ExMsgDesc[31:28]</b>
	127:124	<b>Reserved</b>	
		Exists If:	([ExDesc.IsReg]==true)
		Format:	MBZ
	123:122	<b>Desc[31:30]</b>	
		Exists If:	([Desc.IsReg]==false)
		Format:	<b>MsgDesc[31:30]</b>
	123:113	<b>Reserved</b>	
		Exists If:	([Desc.IsReg]==true)
		Format:	MBZ
	121:113	<b>Desc[19:11]</b>	
		Exists If:	([Desc.IsReg]==false)
Format:		<b>MsgDesc[19:11]</b>	
112	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
111:104	<b>Src1.RegNum</b>		
	Format:	<b>DirectOperand[13:6]</b>	
103:99	<b>Src1.Length</b>		
	Exists If:	([ExDesc.IsReg]==false)	
	Format:	<b>ExMsgDesc[10:6]</b>	
103:99	<b>Reserved</b>		
	Exists If:	([ExDesc.IsReg]==true)	
	Format:	MBZ	
98	<b>Src1.RegFile</b>		
	Format:	<b>DirectOperand[0]</b>	

## EU\_INSTRUCTION\_SEND

97:96	<b>Reserved</b>	
	Exists If:	(([ExDesc.IsReg]==true)
	Format:	MBZ
97:96	<b>ExDesc[27:26]</b>	
	Exists If:	(([ExDesc.IsReg]==false)
	Format:	<b>ExMsgDesc[27:26]</b>
95:92	<b>SFID</b>	
	Format:	<b>SFID</b>
91:81	<b>Reserved</b>	
	Exists If:	(([Desc.IsReg]==true)
	Format:	MBZ
91:81	<b>Desc[10:0]</b>	
	Exists If:	(([Desc.IsReg]==false)
	Format:	<b>MsgDesc[10:0]</b>
80	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
79:72	<b>Src0.RegNum</b>	
	Format:	<b>DirectOperand[13:6]</b>
71	<b>MsgDesc[29]</b>	
	Exists If:	(([Desc.IsReg]==false)
	Format:	<b>MsgDesc[29:29]</b>
71:67	<b>Reserved</b>	
	Exists If:	(([Desc.IsReg]==true)
	Format:	MBZ
70:67	<b>Src0.Length</b>	
	Exists If:	(([Desc.IsReg]==false)
	Format:	<b>MsgDesc[28:25]</b>
66	<b>Src0.RegFile</b>	
	Format:	<b>DirectOperand[0]</b>
65:64	<b>Reserved</b>	
	Exists If:	(([ExDesc.IsReg]==true)
	Format:	MBZ
65:64	<b>ExDesc[25:24]</b>	
	Exists If:	(([ExDesc.IsReg]==false)
	Format:	<b>ExMsgDesc[25:24]</b>

<b>EU_INSTRUCTION_SEND</b>		
63:56	<b>Dst.RegNum</b>	
	Format: <b>DirectOperand[13:6]</b>	
55:51	<b>Reserved</b>	
	Exists If: ([Desc.IsReg]==true) Format: MBZ	
55:51	<b>Dst.Length</b>	
	Exists If: ([Desc.IsReg]==false) Format: <b>MsgDesc[24:20]</b>	
50	<b>Dst.RegFile</b>	
	Format: <b>DirectOperand[0]</b>	
49	<b>ExDesc.IsReg</b> This field indicates that the extended message descriptor is provided by the address register, selected by the AddrSubRegNum[3:1].	
	<b>Value</b>	<b>Name</b>
	0	false
	1	true
48	<b>Desc.IsReg</b> This field indicates that the message descriptor is provided by the address subregister a0.0.	
	<b>Value</b>	<b>Name</b>
	0	false
	1	true
47:43	<b>Reserved</b>	
	Exists If: ([ExDesc.IsReg]==true) Format: MBZ	
47:35	<b>ExDesc[23:11]</b>	
	Exists If: ([ExDesc.IsReg]==false) Format: <b>ExMsgDesc[23:11]</b>	
42:40	<b>AddrSubRegNum[3:1]</b>	
	Exists If: ([ExDesc.IsReg]==true) Format: <b>AddrSubRegNum[3:1]</b>	
38:36	<b>Reserved</b>	
	Exists If: ([ExDesc.IsReg]==true) Format: MBZ	
35	<b>Reserved</b>	
	Exists If: ([ExDesc.IsReg]==true) Format: MBZ	

## EU\_INSTRUCTION\_SEND

34	<b>EOT</b> 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.		
	<b>Value</b>	<b>Name</b>	
	0	Thread is not terminated	
	1	EOT	
33	<b>FusionCtrl</b> This field provides explicit control for EU fusion lock-step execution. When this bit is set to 1b, the instruction is executed serially starting from the first EU to the last EU in the fused set.		
	<b>Value</b>	<b>Name</b>	
	0	Normal lockstep execution	
	1	Serialized execution	
32	<b>AtomicCtrl</b> Format: <span style="float: right;"><b>AtomicCtrl</b></span>		
31	<b>MaskCtrl</b> Mask Control (formerly Write Enable Control). This field determines if the per channel write enables are used to generate the final write enable. This field should be normally "0".		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0	Normal <b>[Default]</b>	Normal. Per channel write enable used for final write enable generation.
	1	NoMask	NoMask.Skips the check for PciP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.
30	<b>Reserved</b>		
29	<b>CmptCtrl</b> Format: <span style="float: right;">MBZ</span> 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, the variations supported by those lookup tables and the compact format. See EU Compact Instruction Format for more information.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0	NoCompaction <b>[Default]</b>	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\_SEND

28	<p><b>PredInv</b></p> <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" 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>Positive <b>[Default]</b></td> <td>Positive polarity of predication. Use the predication mask produced by PredCtrl.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Negative</td> <td>Negative polarity of predication. If PredCtrl is nonzero, invert the predication mask.</td> </tr> </tbody> </table>	Value	Name	Description	0	Positive <b>[Default]</b>	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.
Value	Name	Description								
0	Positive <b>[Default]</b>	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.								
27:24	<p><b>PredCtrl</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>PredCtrl</b></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.</p>	Format:	<b>PredCtrl</b>							
Format:	<b>PredCtrl</b>									
23	<p><b>FlagRegNum[0]</b></p> <p>This field specifies bit[0] of the register number for a flag register operand.</p>									
22	<p><b>FlagSubRegNum</b></p> <p>This field specifies the sub-register number for a flag register operand. There are two sub-registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.</p>									
21:19	<p><b>ChanOff</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>ChanOff</b></td> </tr> </table> <p>This field provides offset information for ARF selection. The can be thought of as a starting channel offset for the execution mask and other ARF registers implicitly accessed.</p>	Format:	<b>ChanOff</b>							
Format:	<b>ChanOff</b>									
18:16	<p><b>ExecSize</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>ExecSize</b></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:	<b>ExecSize</b>							
Format:	<b>ExecSize</b>									
15:0	<p><b>Header</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>Header</b></td> </tr> </table>	Format:	<b>Header</b>							
Format:	<b>Header</b>									

## EU\_INSTRUCTION\_SYNC

EU_INSTRUCTION_SYNC							
Source:	Eulsa						
Size (in bits):	128						
Default Value:	0x00000000, 0x00010000, 0x00000000, 0x00000000						
DWord	Bit	Description					
0..3	127:96	<b>Reserved</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==false)</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Exists If:	([Src0.IsImm]==false)	Format:	MBZ	
	Exists If:	([Src0.IsImm]==false)					
	Format:	MBZ					
	127:96	<b>Src0.ImmValue[31:0]</b> <table border="1"> <tr> <td>Exists If:</td> <td>([Src0.IsImm]==true)</td> </tr> </table>	Exists If:	([Src0.IsImm]==true)			
	Exists If:	([Src0.IsImm]==true)					
	95:92	<b>SyncCtrl</b> <table border="1"> <tr> <td>Format:</td> <td><b>SyncFC</b></td> </tr> </table>	Format:	<b>SyncFC</b>			
	Format:	<b>SyncFC</b>					
	91:88	<b>Reserved</b> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ					
	87	<b>Reserved</b> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ					
	86:80	<b>Reserved</b> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
	Format:	MBZ					
	79:66	<b>Reserved</b> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ			
Format:	MBZ						
65:50	<b>Reserved</b> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
49:48	<b>Dst.HorzStride</b> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>01b</td> <td>1 elements <b>[Default]</b></td> </tr> <tr> <td>Others</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	01b	1 elements <b>[Default]</b>	Others	Reserved
Value	Name						
01b	1 elements <b>[Default]</b>						
Others	Reserved						
47	<b>Reserved</b> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						
46	<b>Src0.IsImm</b> This field indicate that Source 0 operand is carrying an immediate value. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>false</td> </tr> <tr> <td>1</td> <td>true</td> </tr> </tbody> </table>	Value	Name	0	false	1	true
Value	Name						
0	false						
1	true						
45:44	<b>Reserved</b> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ				
Format:	MBZ						



## EU\_INSTRUCTION\_SYNC

43:40	<b>Src0.DataType</b>	
	Exists If:	([Src0.IsImm]==true)
	Format:	<b>ImmDataType</b>
43:40	<b>Reserved</b>	
	Exists If:	([Src0.IsImm]==false)
	Format:	MBZ
39:33	<b>Reserved</b>	
	Format:	MBZ
32	<b>AtomicCtrl</b>	
	Format:	<b>AtomicCtrl</b>
31	<b>MaskCtrl</b>	
	Mask Control (formerly Write Enable Control). This field determines if the per channel write enables are used to generate the final write enable. This field should be normally "0".	
	<b>Value</b>	<b>Name</b>
	<b>Description</b>	
0	Normal <b>[Default]</b>	Normal. Per channel write enable used for final write enable generation.
1	NoMask	NoMask.Skips the check for PclP[n] == ExIP before enabling a channel, as described in the Evaluate Write Enable section.
30	<b>Reserved</b>	
29	<b>CmptCtrl</b>	
	Format:	MBZ
	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, the variations supported by those lookup tables and the compact format. See EU Compact Instruction Format for more information.	
	<b>Value</b>	<b>Name</b>
<b>Description</b>		
0	NoCompaction <b>[Default]</b>	No compaction. 128-bit native instruction supporting all instruction options.
1	Compacted	Compaction is enabled. 64-bit compact instruction supporting only some instruction variations.
28	<b>PredInv</b>	
	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	
	<b>Value</b>	<b>Name</b>
	<b>Description</b>	
0	Positive	Positive polarity of predication. Use the predication mask produced

<b>EU_INSTRUCTION_SYNC</b>								
		<table border="1"> <tr> <td></td> <td><b>[Default]</b></td> <td>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> </table>		<b>[Default]</b>	by PredCtrl.	1	Negative	Negative polarity of predication. If PredCtrl is nonzero, invert the predication mask.
	<b>[Default]</b>	by PredCtrl.						
1	Negative	Negative polarity of predication. If PredCtrl is nonzero, invert the predication mask.						
27:24	<b>PredCtrl</b>	<table border="1"> <tr> <td>Format:</td> <td><b>PredCtrl</b></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.</p>	Format:	<b>PredCtrl</b>				
Format:	<b>PredCtrl</b>							
23	<b>FlagRegNum[0]</b>	This field specifies bit[0] of the register number for a flag register operand.						
22	<b>FlagSubRegNum</b>	This field specifies the sub-register number for a flag register operand. There are two sub-registers in the flag register. Each sub-register contains 16 flag bits. The selected flag sub-register is the source for predication if predication is enabled for the instruction. It is the destination to store conditional flag bits if conditional modifier is enabled for the instruction. The same flag sub-register can be both the predication source and conditional destination, if both predication and conditional modifier are enabled.						
21:19	<b>ChanOff</b>	<table border="1"> <tr> <td>Format:</td> <td><b>ChanOff</b></td> </tr> </table> <p>This field provides offset information for ARF selection. The can be thought of as a starting channel offset for the execution mask and other ARF registers implicitly accessed.</p>	Format:	<b>ChanOff</b>				
Format:	<b>ChanOff</b>							
18:16	<b>ExecSize</b>	<table border="1"> <tr> <td>Format:</td> <td><b>ExecSize</b></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:	<b>ExecSize</b>				
Format:	<b>ExecSize</b>							
15:0	<b>Header</b>	<table border="1"> <tr> <td>Format:</td> <td><b>Header</b></td> </tr> </table>	Format:	<b>Header</b>				
Format:	<b>Header</b>							

## Event Data Payload

<b>MDP_EVENT - Event Data Payload</b>		
Source:	EuSubFunctionGateway	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:24	<b>Reserved</b>
		Access: RO
		Format: MBZ
	23:0	<b>Event ID</b>
		Format: U24
		Indicates the ID of the event to be signalled.
1..7	223:0	<b>Reserved</b>
		Access: RO
		Format: MBZ

## ExMsgDesc

<b>ExMsgDesc</b>		
Size (in bits):		32
Default Value:		0x00000000
DWord	Bit	Description
0	31:12	<b>ExtendedFunctionControl[31:12]</b> Format: <span style="float: right;">U20</span> This field is sent to the target function unit as extended function control.
	11	<b>CPS LOD Compensation</b>
	10:6	<b>Extended Message Length</b> Format: <span style="float: right;">U5</span> This field specifies the number of 256-bit GRF registers starting from <Src1.RegNum> to be sent out on the request message payload <div style="text-align: center; background-color: #e6f2ff; padding: 2px;"><b>Programming Notes</b></div> When <Src1> is null this field must be 0.
	5:0	<b>Reserved</b> Format: <span style="float: right;">MBZ</span>

## Extended Message Descriptor Render Target

Extended Message Descriptor Render Target			
Size (in bits):	32		
Default Value:	0x00000000		
DWord	Bit	Description	
0	31:25	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	24:21	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	20	<b>Null Render Target</b>	
		<b>Description</b>	
		When this bit is set, RT write or read message is considered to be a dummy message and as if it is directed to the NULL render target. Setting this bit in the descriptor, allows SW to not use any entry from the Binding Table to convey NULL RT.	
		<b>Programming Notes</b>	
		SW must set this bit for Render Target Write just to clear the Pixel Scoreboard without allocating an entry in the Binding Table.	
		For Texel Shaders, this bit must be set while sending a NULL Render Target Write Message to communicate End of Texel Shader Thread (EOT) to the AMFSunit.	
19:16	<b>Pixel shading phase for CPS+PS inner loop</b>		
	Format:	U4	
	The loop counter value of a PS phase within CPS+PS(+S) monolithic shader; this value is same as value delivered to Pixel Interpolator when requesting input data for a new PS loop phase. Data Port uses this index to match pixel XY positions delivered by bypass path from PI hardware when a new phase started.		
	<b>Programming Notes</b>		
	The SIMD width of a render target read/write message with PS phase counter must match SIMD width of the Pixel Interpolator Pull message which returns PS phase counter.		
15	<b>Src0 Alpha Present</b>		
	<b>Description</b>		
	Setting this bit indicates that Src0 Alpha is present in the Render Target Write Message.		
	<b>Programming Notes</b>		
	SW must not send a header to send Src0 Alpha present, but instead, it must set this bit and avoid sending the header for RT write messages.		

<b>Extended Message Descriptor Render Target</b>						
14:12	<b>Render Target Array Index</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Description</b></td> </tr> </table> <p>This bit-field is used to set the Render Target Index for MRT messages.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>SW must not send a header to send Render Target Array Index, but instead, it must set this bit-field appropriately and avoid sending the header for RT messages.</p>	Format:	U3	<b>Description</b>	<b>Programming Notes</b>	
	Format:	U3				
	<b>Description</b>					
	<b>Programming Notes</b>					
	11:10	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
		Access:	RO			
Format:	MBZ					
9:6	<b>Extended Message Length</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the number of 256-bit GRF registers starting from &lt;src1&gt; to be sent out on the request message payload. Valid value ranges from 0 to 15. Must be 0 when &lt;src1&gt; is null register.</p>	Format:	U4			
	Format:	U4				
5	<b>End of Thread</b> This field, if set, indicates that this is the final message of the thread and the threads resources can be reclaimed.					
4	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO				
Format:	MBZ					
3:0	<b>Target Function ID</b> 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		
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:12	<b>Bindless Surface Offset</b>
		Format: SurfaceStateOffset[25:6] Specifies the bindless surface offset if the <b>Binding Table Index</b> is set to 252. Ignored otherwise. The bindless surface offset is added to the <b>Bindless Surface Base Address</b> as bits 25:6 of the byte-based address. The resulting address is the location of SURFACE_STATE for this message.
	11	<b>CPS Message LOD Compensation Enable</b>
		Format: Enable Specifies whether LOD Compensation is enabled for this message. See <b>CPS LOD Compensation Enable</b> in SAMPLER_STATE for more details.
		<b>Programming Notes</b>
		This field must be disabled if the response length of the message is zero.
		This field must be disabled if the messages is from a 32-pixel dispatch thread.
		This field must be disabled unless <b>SIMD Mode</b> is SIMD8* or SIMD16*.
10:0		<b>Execution Unit Extended Message Descriptor Definition</b>
		Format: ExMsgDesc[10:0]



## FFTID Message Header Control

MHC_FFTID - FFTID Message Header Control			
Size (in bits):		32	
Default Value:		0x00000000	
DWord	Bit	Description	
0	31:8	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	7:0	<b>FFTID</b>	
	Format:	U8	
Fixed function thread ID, used to free up resources by the thread on thread completion.			



## Filter\_Coefficient

Filter_Coefficient		
Size (in bits):	8	
Default Value:	0x00000000	
DWord	Bit	Description
0	7:0	<b>Filter Coefficient</b> Format: S1.6 Range : [-1 63/64, +1 63/64]

## Filter\_Coefficients

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

## FrameDeltaQp

FrameDeltaQp		
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:56	<b>FrameDeltaQp[7]</b> Format: S7
	55:48	<b>FrameDeltaQp[6]</b> Format: S7
	47:40	<b>FrameDeltaQp[5]</b> Format: S7
	39:32	<b>FrameDeltaQp[4]</b> Format: S7
	31:24	<b>FrameDeltaQp[3]</b> Format: S7
	23:16	<b>FrameDeltaQp[2]</b> Format: S7
	15:8	<b>FrameDeltaQp[1]</b> Format: S7
	7:0	<b>FrameDeltaQp[0]</b> Format: S7

## FrameDeltaQpRange

<b>FrameDeltaQpRange</b>		
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:56	<b>FrameDeltaQpRange[7]</b> Format: <span style="float: right;">U8</span>
	55:48	<b>FrameDeltaQpRange[6]</b> Format: <span style="float: right;">U8</span>
	47:40	<b>FrameDeltaQpRange[5]</b> Format: <span style="float: right;">U8</span>
	39:32	<b>FrameDeltaQpRange[4]</b> Format: <span style="float: right;">U8</span>
	31:24	<b>FrameDeltaQpRange[3]</b> Format: <span style="float: right;">U8</span>
	23:16	<b>FrameDeltaQpRange[2]</b> Format: <span style="float: right;">U8</span>
	15:8	<b>FrameDeltaQpRange[1]</b> Format: <span style="float: right;">U8</span>
	7:0	<b>FrameDeltaQpRange[0]</b> Format: <span style="float: right;">U8</span>

## Gamut\_Expansion\_Gamma\_Correction

<b>Gamut_Expansion_Gamma_Correction</b>	
Source:	VideoEnhancementCS
Size (in bits):	32768
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x01000100, 0x01000100, 0x01000100, 0x01000100, 0x02000200, 0x02000200, 0x02000200, 0x02000200, 0x03000300, 0x03000300, 0x03000300, 0x03000300, 0x04000400, 0x04000400, 0x04000400, 0x04000400, 0x05000500, 0x05000500, 0x05000500, 0x05000500, 0x06000600, 0x06000600, 0x06000600, 0x06000600, 0x07000700, 0x07000700, 0x07000700, 0x07000700, 0x08000800, 0x08000800, 0x08000800, 0x08000800, 0x09000900, 0x09000900, 0x09000900, 0x09000900, 0x0A000A00, 0x0A000A00, 0x0A000A00, 0x0A000A00, 0x0B000B00, 0x0B000B00, 0x0B000B00, 0x0B000B00, 0x0C000C00, 0x0C000C00, 0x0C000C00, 0x0C000C00, 0x0D000D00, 0x0D000D00, 0x0D000D00, 0x0D000D00, 0x0E000E00, 0x0E000E00, 0x0E000E00, 0x0E000E00, 0x0F000F00, 0x0F000F00, 0x0F000F00, 0x0F000F00, 0x10001000, 0x10001000, 0x10001000, 0x10001000, 0x11001100, 0x11001100, 0x11001100, 0x11001100, 0x12001200, 0x12001200, 0x12001200, 0x12001200, 0x13001300, 0x13001300, 0x13001300, 0x13001300, 0x14001400, 0x14001400, 0x14001400, 0x14001400, 0x15001500, 0x15001500, 0x15001500, 0x15001500, 0x16001600, 0x16001600, 0x16001600, 0x16001600, 0x17001700, 0x17001700, 0x17001700, 0x17001700, 0x18001800, 0x18001800, 0x18001800, 0x18001800, 0x19001900, 0x19001900, 0x19001900, 0x19001900, 0x1A001A00, 0x1A001A00, 0x1A001A00, 0x1A001A00, 0x1B001B00, 0x1B001B00, 0x1B001B00, 0x1B001B00, 0x1C001C00, 0x1C001C00, 0x1C001C00, 0x1C001C00, 0x1D001D00, 0x1D001D00, 0x1D001D00, 0x1D001D00, 0x1E001E00, 0x1E001E00, 0x1E001E00, 0x1E001E00, 0x1F001F00, 0x1F001F00, 0x1F001F00, 0x1F001F00, 0x20002000, 0x20002000, 0x20002000, 0x20002000, 0x21002100, 0x21002100, 0x21002100, 0x21002100, 0x22002200, 0x22002200, 0x22002200, 0x22002200, 0x23002300, 0x23002300, 0x23002300, 0x23002300, 0x24002400, 0x24002400, 0x24002400, 0x24002400, 0x25002500, 0x25002500, 0x25002500, 0x25002500, 0x26002600, 0x26002600, 0x26002600, 0x26002600, 0x27002700, 0x27002700, 0x27002700, 0x27002700, 0x28002800, 0x28002800, 0x28002800, 0x28002800, 0x29002900, 0x29002900, 0x29002900, 0x29002900, 0x2A002A00, 0x2A002A00, 0x2A002A00, 0x2A002A00, 0x2B002B00, 0x2B002B00, 0x2B002B00, 0x2B002B00, 0x2C002C00, 0x2C002C00, 0x2C002C00, 0x2C002C00, 0x2D002D00, 0x2D002D00, 0x2D002D00, 0x2D002D00, 0x2E002E00, 0x2E002E00, 0x2E002E00, 0x2E002E00, 0x2F002F00, 0x2F002F00, 0x2F002F00, 0x2F002F00, 0x30003000, 0x30003000, 0x30003000, 0x30003000, 0x31003100, 0x31003100, 0x31003100, 0x31003100, 0x32003200, 0x32003200, 0x32003200, 0x32003200, 0x32003200, 0x33003300, 0x33003300, 0x33003300, 0x33003300, 0x34003400, 0x34003400, 0x34003400, 0x34003400, 0x35003500, 0x35003500, 0x35003500, 0x35003500, 0x36003600, 0x36003600, 0x36003600, 0x36003600, 0x37003700, 0x37003700, 0x37003700, 0x37003700, 0x38003800, 0x38003800, 0x38003800, 0x38003800, 0x39003900, 0x39003900, 0x39003900, 0x39003900, 0x3A003A00, 0x3A003A00, 0x3A003A00, 0x3A003A00, 0x3B003B00, 0x3B003B00, 0x3B003B00, 0x3B003B00, 0x3C003C00, 0x3C003C00, 0x3C003C00, 0x3C003C00, 0x3D003D00, 0x3D003D00, 0x3D003D00,





## Gamut\_Expansion\_Gamma\_Correction

0xC800C800, 0xC800C800, 0xC900C900, 0xC900C900, 0xC900C900, 0xC900C900,  
 0xCA00CA00, 0xCA00CA00, 0xCA00CA00, 0xCA00CA00, 0xCB00CB00, 0xCB00CB00,  
 0xCB00CB00, 0xCB00CB00, 0xCC00CC00, 0xCC00CC00, 0xCC00CC00, 0xCC00CC00,  
 0xCD00CD00, 0xCD00CD00, 0xCD00CD00, 0xCD00CD00, 0xCE00CE00, 0xCE00CE00,  
 0xCE00CE00, 0xCE00CE00, 0xCF00CF00, 0xCF00CF00, 0xCF00CF00, 0xCF00CF00,  
 0xD000D000, 0xD000D000, 0xD000D000, 0xD000D000, 0xD100D100, 0xD100D100,  
 0xD100D100, 0xD100D100, 0xD200D200, 0xD200D200, 0xD200D200, 0xD200D200,  
 0xD300D300, 0xD300D300, 0xD300D300, 0xD300D300, 0xD400D400, 0xD400D400,  
 0xD400D400, 0xD400D400, 0xD500D500, 0xD500D500, 0xD500D500, 0xD500D500,  
 0xD600D600, 0xD600D600, 0xD600D600, 0xD600D600, 0xD700D700, 0xD700D700,  
 0xD700D700, 0xD700D700, 0xD800D800, 0xD800D800, 0xD800D800, 0xD800D800,  
 0xD900D900, 0xD900D900, 0xD900D900, 0xD900D900, 0xDA00DA00, 0xDA00DA00,  
 0xDA00DA00, 0xDA00DA00, 0xDB00DB00, 0xDB00DB00, 0xDB00DB00, 0xDB00DB00,  
 0xDC00DC00, 0xDC00DC00, 0xDC00DC00, 0xDC00DC00, 0xDD00DD00, 0xDD00DD00,  
 0xDD00DD00, 0xDD00DD00, 0xDE00DE00, 0xDE00DE00, 0xDE00DE00, 0xDE00DE00,  
 0xDF00DF00, 0xDF00DF00, 0xDF00DF00, 0xDF00DF00, 0xE000E000, 0xE000E000,  
 0xE000E000, 0xE000E000, 0xE100E100, 0xE100E100, 0xE100E100, 0xE100E100,  
 0xE200E200, 0xE200E200, 0xE200E200, 0xE200E200, 0xE300E300, 0xE300E300,  
 0xE300E300, 0xE300E300, 0xE400E400, 0xE400E400, 0xE400E400, 0xE400E400,  
 0xE500E500, 0xE500E500, 0xE500E500, 0xE500E500, 0xE600E600, 0xE600E600,  
 0xE600E600, 0xE600E600, 0xE700E700, 0xE700E700, 0xE700E700, 0xE700E700,  
 0xE800E800, 0xE800E800, 0xE800E800, 0xE800E800, 0xE900E900, 0xE900E900,  
 0xE900E900, 0xE900E900, 0xEA00EA00, 0xEA00EA00, 0xEA00EA00, 0xEA00EA00,  
 0xEB00EB00, 0xEB00EB00, 0xEB00EB00, 0xEB00EB00, 0xEC00EC00, 0xEC00EC00,  
 0xEC00EC00, 0xEC00EC00, 0xED00ED00, 0xED00ED00, 0xED00ED00, 0xED00ED00,  
 0xEE00EE00, 0xEE00EE00, 0xEE00EE00, 0xEE00EE00, 0xEF00EF00, 0xEF00EF00, 0xEF00EF00,  
 0xEF00EF00, 0xF000F000, 0xF000F000, 0xF000F000, 0xF000F000, 0xF100F100, 0xF100F100,  
 0xF100F100, 0xF100F100, 0xF200F200, 0xF200F200, 0xF200F200, 0xF200F200, 0xF300F300,  
 0xF300F300, 0xF300F300, 0xF300F300, 0xF400F400, 0xF400F400, 0xF400F400, 0xF400F400,  
 0xF500F500, 0xF500F500, 0xF500F500, 0xF500F500, 0xF600F600, 0xF600F600, 0xF600F600,  
 0xF600F600, 0xF700F700, 0xF700F700, 0xF700F700, 0xF700F700, 0xF800F800, 0xF800F800,  
 0xF800F800, 0xF800F800, 0xF900F900, 0xF900F900, 0xF900F900, 0xF900F900,  
 0xFA00FA00, 0xFA00FA00, 0xFA00FA00, 0xFA00FA00, 0xFB00FB00, 0xFB00FB00,  
 0xFB00FB00, 0xFB00FB00, 0xFC00FC00, 0xFC00FC00, 0xFC00FC00, 0xFC00FC00,  
 0xFD00FD00, 0xFD00FD00, 0xFD00FD00, 0xFD00FD00, 0xFE00FE00, 0xFE00FE00,  
 0xFE00FE00, 0xFE00FE00, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF

### Programming Notes

The default values follow the pattern suggested by incomplete table below.

DWords	DWord 0	DWord 1	DWord 2	DWord 3
0..3 : Point[0]	00000000h	00000000h	00000000h	00000000h
4..7 : Point[1]	01000100h	01000100h	01000100h	01000100h
8..11 : Point[2]	02000200h	02000200h	02000200h	02000200h
12..15 : Point[3]	03000300h	03000300h	03000300h	03000300h

...



## Gamut\_Expansion\_Gamma\_Correction

1016..1019 : Point[254]	fe00fe00h	fe00fe00h	fe00fe00h	fe00fe00h
1020..1023 : Point[255]	ffffffffh	ffffffffh	ffffffffh	ffffffffh

DWord	Bit	Description
0..1	63:48	<b>Inverse R-ch Gamma Corrected Value 0</b>
		Default Value: 0000h
		Format: U16
	47:32	<b>Inverse Pixel Value 0</b>
		Default Value: 0000h
		Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 0</b>
		Default Value: 0000h
		Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 0</b>
		Default Value: 0000h
		Format: U16
2..3	63:48	<b>Forward R-ch Gamma Corrected Value 0</b>
		Default Value: 0000h
		Format: U16
	47:32	<b>Forward Pixel Value 0</b>
		Default Value: 0000h
		Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 0</b>
		Default Value: 0000h
		Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 0</b>
		Default Value: 0000h
		Format: U16
4..5	63:48	<b>Inverse R-ch Gamma Corrected Value 1</b>
		Default Value: 0100h
		Format: U16
	47:32	<b>Inverse Pixel Value 1</b>
		Default Value: 0100h
		Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 1</b>
		Default Value: 0100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 1</b>
		Default Value: 0100h Format: U16
6..7	63:48	<b>Forward R-ch Gamma Corrected Value 1</b>
		Default Value: 0100h Format: U16
	47:32	<b>Forward Pixel Value 1</b>
		Default Value: 0100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 1</b>
		Default Value: 0100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 1</b>
		Default Value: 0100h Format: U16
8..9	63:48	<b>Inverse R-ch Gamma Corrected Value 2</b>
		Default Value: 0200h Format: U16
	47:32	<b>Inverse Pixel Value 2</b>
		Default Value: 0200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 2</b>
		Default Value: 0200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 2</b>
		Default Value: 0200h Format: U16
10..11	63:48	<b>Forward R-ch Gamma Corrected Value 2</b>
		Default Value: 0200h Format: U16
	47:32	<b>Forward Pixel Value 2</b>
		Default Value: 0200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 2</b>
		Default Value: 0200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 2</b>
		Default Value: 0200h Format: U16
12..13	63:48	<b>Inverse R-ch Gamma Corrected Value 3</b>
		Default Value: 0300h Format: U16
	47:32	<b>Inverse Pixel Value 3</b>
		Default Value: 0300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 3</b>
		Default Value: 0300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 3</b>
		Default Value: 0300h Format: U16
14..15	63:48	<b>Forward R-ch Gamma Corrected Value 3</b>
		Default Value: 0300h Format: U16
	47:32	<b>Forward Pixel Value 3</b>
		Default Value: 0300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 3</b>
		Default Value: 0300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 3</b>
		Default Value: 0300h Format: U16
16..17	63:48	<b>Inverse R-ch Gamma Corrected Value 4</b>
		Default Value: 0400h Format: U16
	47:32	<b>Inverse Pixel Value 4</b>
		Default Value: 0400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 4</b>
		Default Value: 0400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 4</b>
		Default Value: 0400h Format: U16
18..19	63:48	<b>Forward R-ch Gamma Corrected Value 4</b>
		Default Value: 0400h Format: U16
	47:32	<b>Forward Pixel Value 4</b>
		Default Value: 0400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 4</b>
		Default Value: 0400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 4</b>
		Default Value: 0400h Format: U16
20..21	63:48	<b>Inverse R-ch Gamma Corrected Value 5</b>
		Default Value: 0500h Format: U16
	47:32	<b>Inverse Pixel Value 5</b>
		Default Value: 0500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 5</b>
		Default Value: 0500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 5</b>
		Default Value: 0500h Format: U16
22..23	63:48	<b>Forward R-ch Gamma Corrected Value 5</b>
		Default Value: 0500h Format: U16
	47:32	<b>Forward Pixel Value 5</b>
		Default Value: 0500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 5</b>
		Default Value: 0500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 5</b>
		Default Value: 0500h Format: U16
24..25	63:48	<b>Inverse R-ch Gamma Corrected Value 6</b>
		Default Value: 0600h Format: U16
	47:32	<b>Inverse Pixel Value 6</b>
		Default Value: 0600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 6</b>
		Default Value: 0600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 6</b>
		Default Value: 0600h Format: U16
26..27	63:48	<b>Forward R-ch Gamma Corrected Value 6</b>
		Default Value: 0600h Format: U16
	47:32	<b>Forward Pixel Value 6</b>
		Default Value: 0600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 6</b>
		Default Value: 0600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 6</b>
		Default Value: 0600h Format: U16
28..29	63:48	<b>Inverse R-ch Gamma Corrected Value 7</b>
		Default Value: 0700h Format: U16
	47:32	<b>Inverse Pixel Value 7</b>
		Default Value: 0700h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 7</b>
		Default Value: 0700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 7</b>
		Default Value: 0700h Format: U16
30..31	63:48	<b>Forward R-ch Gamma Corrected Value 7</b>
		Default Value: 0700h Format: U16
	47:32	<b>Forward Pixel Value 7</b>
		Default Value: 0700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 7</b>
		Default Value: 0700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 7</b>
		Default Value: 0700h Format: U16
32..33	63:48	<b>Inverse R-ch Gamma Corrected Value 8</b>
		Default Value: 0800h Format: U16
	47:32	<b>Inverse Pixel Value 8</b>
		Default Value: 0800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 8</b>
		Default Value: 0800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 8</b>
		Default Value: 0800h Format: U16
34..35	63:48	<b>Forward R-ch Gamma Corrected Value 8</b>
		Default Value: 0800h Format: U16
	47:32	<b>Forward Pixel Value 8</b>
		Default Value: 0800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 8</b>
		Default Value: 0800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 8</b>
		Default Value: 0800h Format: U16
36..37	63:48	<b>Inverse R-ch Gamma Corrected Value 9</b>
		Default Value: 0900h Format: U16
	47:32	<b>Inverse Pixel Value 9</b>
		Default Value: 0900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 9</b>
		Default Value: 0900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 9</b>
		Default Value: 0900h Format: U16
38..39	63:48	<b>Forward R-ch Gamma Corrected Value 9</b>
		Default Value: 0900h Format: U16
	47:32	<b>Forward Pixel Value 9</b>
		Default Value: 0900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 9</b>
		Default Value: 0900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 9</b>
		Default Value: 0900h Format: U16
40..41	63:48	<b>Inverse R-ch Gamma Corrected Value 10</b>
		Default Value: 0a00h Format: U16
	47:32	<b>Inverse Pixel Value 10</b>
		Default Value: 0a00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 10</b>
		Default Value: 0a00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 10</b>
		Default Value: 0a00h Format: U16
42..43	63:48	<b>Forward R-ch Gamma Corrected Value 10</b>
		Default Value: 0a00h Format: U16
	47:32	<b>Forward Pixel Value 10</b>
		Default Value: 0a00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 10</b>
		Default Value: 0a00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 10</b>
		Default Value: 0a00h Format: U16
44..45	63:48	<b>Inverse R-ch Gamma Corrected Value 11</b>
		Default Value: 0b00h Format: U16
	47:32	<b>Inverse Pixel Value 11</b>
		Default Value: 0b00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 11</b>
		Default Value: 0b00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 11</b>
		Default Value: 0b00h Format: U16
46..47	63:48	<b>Forward R-ch Gamma Corrected Value 11</b>
		Default Value: 0b00h Format: U16
	47:32	<b>Forward Pixel Value 11</b>
		Default Value: 0b00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 11</b>
		Default Value: 0b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 11</b>
		Default Value: 0b00h Format: U16
48..49	63:48	<b>Inverse R-ch Gamma Corrected Value 12</b>
		Default Value: 0c00h Format: U16
	47:32	<b>Inverse Pixel Value 12</b>
		Default Value: 0c00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 12</b>
		Default Value: 0c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 12</b>
		Default Value: 0c00h Format: U16
50..51	63:48	<b>Forward R-ch Gamma Corrected Value 12</b>
		Default Value: 0c00h Format: U16
	47:32	<b>Forward Pixel Value 12</b>
		Default Value: 0c00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 12</b>
		Default Value: 0c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 12</b>
		Default Value: 0c00h Format: U16
52..53	63:48	<b>Inverse R-ch Gamma Corrected Value 13</b>
		Default Value: 0d00h Format: U16
	47:32	<b>Inverse Pixel Value 13</b>
		Default Value: 0d00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 13</b>
		Default Value: 0d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 13</b>
		Default Value: 0d00h Format: U16
54..55	63:48	<b>Forward R-ch Gamma Corrected Value 13</b>
		Default Value: 0d00h Format: U16
	47:32	<b>Forward Pixel Value 13</b>
		Default Value: 0d00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 13</b>
		Default Value: 0d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 13</b>
		Default Value: 0d00h Format: U16
56..57	63:48	<b>Inverse R-ch Gamma Corrected Value 14</b>
		Default Value: 0e00h Format: U16
	47:32	<b>Inverse Pixel Value 14</b>
		Default Value: 0e00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 14</b>
		Default Value: 0e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 14</b>
		Default Value: 0e00h Format: U16
58..59	63:48	<b>Forward R-ch Gamma Corrected Value 14</b>
		Default Value: 0e00h Format: U16
	47:32	<b>Forward Pixel Value 14</b>
		Default Value: 0e00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 14</b>
		Default Value: 0e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 14</b>
		Default Value: 0e00h Format: U16
60..61	63:48	<b>Inverse R-ch Gamma Corrected Value 15</b>
		Default Value: 0f00h Format: U16
	47:32	<b>Inverse Pixel Value 15</b>
		Default Value: 0f00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 15</b>
		Default Value: 0f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 15</b>
		Default Value: 0f00h Format: U16
62..63	63:48	<b>Forward R-ch Gamma Corrected Value 15</b>
		Default Value: 0f00h Format: U16
	47:32	<b>Forward Pixel Value 15</b>
		Default Value: 0f00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 15</b>
		Default Value: 0f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 15</b>
		Default Value: 0f00h Format: U16
64..65	63:48	<b>Inverse R-ch Gamma Corrected Value 16</b>
		Default Value: 1000h Format: U16
	47:32	<b>Inverse Pixel Value 16</b>
		Default Value: 1000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 16</b>
		Default Value: 1000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 16</b>
		Default Value: 1000h Format: U16
66..67	63:48	<b>Forward R-ch Gamma Corrected Value 16</b>
		Default Value: 1000h Format: U16
	47:32	<b>Forward Pixel Value 16</b>
		Default Value: 1000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 16</b>
		Default Value: 1000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 16</b>
		Default Value: 1000h Format: U16
68..69	63:48	<b>Inverse R-ch Gamma Corrected Value 17</b>
		Default Value: 1100h Format: U16
	47:32	<b>Inverse Pixel Value 17</b>
		Default Value: 1100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 17</b>
		Default Value: 1100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 17</b>
		Default Value: 1100h Format: U16
70..71	63:48	<b>Forward R-ch Gamma Corrected Value 17</b>
		Default Value: 1100h Format: U16
	47:32	<b>Forward Pixel Value 17</b>
		Default Value: 1100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 17</b>
		Default Value: 1100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 17</b>
		Default Value: 1100h Format: U16
72..73	63:48	<b>Inverse R-ch Gamma Corrected Value 18</b>
		Default Value: 1200h Format: U16
	47:32	<b>Inverse Pixel Value 18</b>
		Default Value: 1200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 18</b>
		Default Value: 1200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 18</b>
		Default Value: 1200h Format: U16
74..75	63:48	<b>Forward R-ch Gamma Corrected Value 18</b>
		Default Value: 1200h Format: U16
	47:32	<b>Forward Pixel Value 18</b>
		Default Value: 1200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 18</b>
		Default Value: 1200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 18</b>
		Default Value: 1200h Format: U16
76..77	63:48	<b>Inverse R-ch Gamma Corrected Value 19</b>
		Default Value: 1300h Format: U16
	47:32	<b>Inverse Pixel Value 19</b>
		Default Value: 1300h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 19</b>
		Default Value: 1300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 19</b>
		Default Value: 1300h Format: U16
78..79	63:48	<b>Forward R-ch Gamma Corrected Value 19</b>
		Default Value: 1300h Format: U16
	47:32	<b>Forward Pixel Value 19</b>
		Default Value: 1300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 19</b>
		Default Value: 1300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 19</b>
		Default Value: 1300h Format: U16
80..81	63:48	<b>Inverse R-ch Gamma Corrected Value 20</b>
		Default Value: 1400h Format: U16
	47:32	<b>Inverse Pixel Value 20</b>
		Default Value: 1400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 20</b>
		Default Value: 1400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 20</b>
		Default Value: 1400h Format: U16
82..83	63:48	<b>Forward R-ch Gamma Corrected Value 20</b>
		Default Value: 1400h Format: U16
	47:32	<b>Forward Pixel Value 20</b>
		Default Value: 1400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 20</b>
		Default Value: 1400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 20</b>
		Default Value: 1400h Format: U16
84..85	63:48	<b>Inverse R-ch Gamma Corrected Value 21</b>
		Default Value: 1500h Format: U16
	47:32	<b>Inverse Pixel Value 21</b>
		Default Value: 1500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 21</b>
		Default Value: 1500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 21</b>
		Default Value: 1500h Format: U16
86..87	63:48	<b>Forward R-ch Gamma Corrected Value 21</b>
		Default Value: 1500h Format: U16
	47:32	<b>Forward Pixel Value 21</b>
		Default Value: 1500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 21</b>
		Default Value: 1500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 21</b>
		Default Value: 1500h Format: U16
88..89	63:48	<b>Inverse R-ch Gamma Corrected Value 22</b>
		Default Value: 1600h Format: U16
	47:32	<b>Inverse Pixel Value 22</b>
		Default Value: 1600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 22</b>
		Default Value: 1600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 22</b>
		Default Value: 1600h Format: U16
90..91	63:48	<b>Forward R-ch Gamma Corrected Value 22</b>
		Default Value: 1600h Format: U16
	47:32	<b>Forward Pixel Value 22</b>
		Default Value: 1600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 22</b>
		Default Value: 1600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 22</b>
		Default Value: 1600h Format: U16
92..93	63:48	<b>Inverse R-ch Gamma Corrected Value 23</b>
		Default Value: 1700h Format: U16
	47:32	<b>Inverse Pixel Value 23</b>
		Default Value: 1700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 23</b>
		Default Value: 1700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 23</b>
		Default Value: 1700h Format: U16
94..95	63:48	<b>Forward R-ch Gamma Corrected Value 23</b>
		Default Value: 1700h Format: U16
	47:32	<b>Forward Pixel Value 23</b>
		Default Value: 1700h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 23</b>
		Default Value: 1700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 23</b>
		Default Value: 1700h Format: U16
96..97	63:48	<b>Inverse R-ch Gamma Corrected Value 24</b>
		Default Value: 1800h Format: U16
	47:32	<b>Inverse Pixel Value 24</b>
		Default Value: 1800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 24</b>
		Default Value: 1800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 24</b>
		Default Value: 1800h Format: U16
98..99	63:48	<b>Forward R-ch Gamma Corrected Value 24</b>
		Default Value: 1800h Format: U16
	47:32	<b>Forward Pixel Value 24</b>
		Default Value: 1800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 24</b>
		Default Value: 1800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 24</b>
		Default Value: 1800h Format: U16
100..101	63:48	<b>Inverse R-ch Gamma Corrected Value 25</b>
		Default Value: 1900h Format: U16
	47:32	<b>Inverse Pixel Value 25</b>
		Default Value: 1900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 25</b>
		Default Value: 1900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 25</b>
		Default Value: 1900h Format: U16
102..103	63:48	<b>Forward R-ch Gamma Corrected Value 25</b>
		Default Value: 1900h Format: U16
	47:32	<b>Forward Pixel Value 25</b>
		Default Value: 1900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 25</b>
		Default Value: 1900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 25</b>
		Default Value: 1900h Format: U16
104..105	63:48	<b>Inverse R-ch Gamma Corrected Value 26</b>
		Default Value: 1a00h Format: U16
	47:32	<b>Inverse Pixel Value 26</b>
		Default Value: 1a00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 26</b>
		Default Value: 1a00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 26</b>
		Default Value: 1a00h Format: U16
106..107	63:48	<b>Forward R-ch Gamma Corrected Value 26</b>
		Default Value: 1a00h Format: U16
	47:32	<b>Forward Pixel Value 26</b>
		Default Value: 1a00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 26</b>
		Default Value: 1a00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 26</b>
		Default Value: 1a00h Format: U16
108..109	63:48	<b>Inverse R-ch Gamma Corrected Value 27</b>
		Default Value: 1b00h Format: U16
	47:32	<b>Inverse Pixel Value 27</b>
		Default Value: 1b00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 27</b>
		Default Value: 1b00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 27</b>
		Default Value: 1b00h Format: U16
110..111	63:48	<b>Forward R-ch Gamma Corrected Value 27</b>
		Default Value: 1b00h Format: U16
	47:32	<b>Forward Pixel Value 27</b>
		Default Value: 1b00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 27</b>
		Default Value: 1b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 27</b>
		Default Value: 1b00h Format: U16
112..113	63:48	<b>Inverse R-ch Gamma Corrected Value 28</b>
		Default Value: 1c00h Format: U16
	47:32	<b>Inverse Pixel Value 28</b>
		Default Value: 1c00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 28</b>
		Default Value: 1c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 28</b>
		Default Value: 1c00h Format: U16
114..115	63:48	<b>Forward R-ch Gamma Corrected Value 28</b>
		Default Value: 1c00h Format: U16
	47:32	<b>Forward Pixel Value 28</b>
		Default Value: 1c00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 28</b>
		Default Value: 1c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 28</b>
		Default Value: 1c00h Format: U16
116..117	63:48	<b>Inverse R-ch Gamma Corrected Value 29</b>
		Default Value: 1d00h Format: U16
	47:32	<b>Inverse Pixel Value 29</b>
		Default Value: 1d00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 29</b>
		Default Value: 1d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 29</b>
		Default Value: 1d00h Format: U16
118..119	63:48	<b>Forward R-ch Gamma Corrected Value 29</b>
		Default Value: 1d00h Format: U16
	47:32	<b>Forward Pixel Value 29</b>
		Default Value: 1d00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 29</b>
		Default Value: 1d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 29</b>
		Default Value: 1d00h Format: U16
120..121	63:48	<b>Inverse R-ch Gamma Corrected Value 30</b>
		Default Value: 1e00h Format: U16
	47:32	<b>Inverse Pixel Value 30</b>
		Default Value: 1e00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 30</b>
		Default Value: 1e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 30</b>
		Default Value: 1e00h Format: U16
122..123	63:48	<b>Forward R-ch Gamma Corrected Value 30</b>
		Default Value: 1e00h Format: U16
	47:32	<b>Forward Pixel Value 30</b>
		Default Value: 1e00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 30</b>
		Default Value: 1e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 30</b>
		Default Value: 1e00h Format: U16
124..125	63:48	<b>Inverse R-ch Gamma Corrected Value 31</b>
		Default Value: 1f00h Format: U16
	47:32	<b>Inverse Pixel Value 31</b>
		Default Value: 1f00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 31</b>
		Default Value: 1f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 31</b>
		Default Value: 1f00h Format: U16
126..127	63:48	<b>Forward R-ch Gamma Corrected Value 31</b>
		Default Value: 1f00h Format: U16
	47:32	<b>Forward Pixel Value 31</b>
		Default Value: 1f00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 31</b>
		Default Value: 1f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 31</b>
		Default Value: 1f00h Format: U16
128..129	63:48	<b>Inverse R-ch Gamma Corrected Value 32</b>
		Default Value: 2000h Format: U16
	47:32	<b>Inverse Pixel Value 32</b>
		Default Value: 2000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 32</b>
		Default Value: 2000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 32</b>
		Default Value: 2000h Format: U16
130..131	63:48	<b>Forward R-ch Gamma Corrected Value 32</b>
		Default Value: 2000h Format: U16
	47:32	<b>Forward Pixel Value 32</b>
		Default Value: 2000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 32</b>
		Default Value: 2000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 32</b>
		Default Value: 2000h Format: U16
132..133	63:48	<b>Inverse R-ch Gamma Corrected Value 33</b>
		Default Value: 2100h Format: U16
	47:32	<b>Inverse Pixel Value 33</b>
		Default Value: 2100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 33</b>
		Default Value: 2100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 33</b>
		Default Value: 2100h Format: U16
134..135	63:48	<b>Forward R-ch Gamma Corrected Value 33</b>
		Default Value: 2100h Format: U16
	47:32	<b>Forward Pixel Value 33</b>
		Default Value: 2100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 33</b>
		Default Value: 2100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 33</b>
		Default Value: 2100h Format: U16
136..137	63:48	<b>Inverse R-ch Gamma Corrected Value 34</b>
		Default Value: 2200h Format: U16
	47:32	<b>Inverse Pixel Value 34</b>
		Default Value: 2200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 34</b>
		Default Value: 2200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 34</b>
		Default Value: 2200h Format: U16
138..139	63:48	<b>Forward R-ch Gamma Corrected Value 34</b>
		Default Value: 2200h Format: U16
	47:32	<b>Forward Pixel Value 34</b>
		Default Value: 2200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 34</b>
		Default Value: 2200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 34</b>
		Default Value: 2200h Format: U16
140..141	63:48	<b>Inverse R-ch Gamma Corrected Value 35</b>
		Default Value: 2300h Format: U16
	47:32	<b>Inverse Pixel Value 35</b>
		Default Value: 2300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 35</b>
		Default Value: 2300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 35</b>
		Default Value: 2300h Format: U16
142..143	63:48	<b>Forward R-ch Gamma Corrected Value 35</b>
		Default Value: 2300h Format: U16
	47:32	<b>Forward Pixel Value 35</b>
		Default Value: 2300h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 35</b>
		Default Value: 2300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 35</b>
		Default Value: 2300h Format: U16
144..145	63:48	<b>Inverse R-ch Gamma Corrected Value 36</b>
		Default Value: 2400h Format: U16
	47:32	<b>Inverse Pixel Value 36</b>
		Default Value: 2400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 36</b>
		Default Value: 2400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 36</b>
		Default Value: 2400h Format: U16
146..147	63:48	<b>Forward R-ch Gamma Corrected Value 36</b>
		Default Value: 2400h Format: U16
	47:32	<b>Forward Pixel Value 36</b>
		Default Value: 2400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 36</b>
		Default Value: 2400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 36</b>
		Default Value: 2400h Format: U16
148..149	63:48	<b>Inverse R-ch Gamma Corrected Value 37</b>
		Default Value: 2500h Format: U16
	47:32	<b>Inverse Pixel Value 37</b>
		Default Value: 2500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 37</b>
		Default Value: 2500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 37</b>
		Default Value: 2500h Format: U16
150..151	63:48	<b>Forward R-ch Gamma Corrected Value 37</b>
		Default Value: 2500h Format: U16
	47:32	<b>Forward Pixel Value 37</b>
		Default Value: 2500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 37</b>
		Default Value: 2500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 37</b>
		Default Value: 2500h Format: U16
152..153	63:48	<b>Inverse R-ch Gamma Corrected Value 38</b>
		Default Value: 2600h Format: U16
	47:32	<b>Inverse Pixel Value 38</b>
		Default Value: 2600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 38</b>
		Default Value: 2600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 38</b>
		Default Value: 2600h Format: U16
154..155	63:48	<b>Forward R-ch Gamma Corrected Value 38</b>
		Default Value: 2600h Format: U16
	47:32	<b>Forward Pixel Value 38</b>
		Default Value: 2600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 38</b>
		Default Value: 2600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 38</b>
		Default Value: 2600h Format: U16
156..157	63:48	<b>Inverse R-ch Gamma Corrected Value 39</b>
		Default Value: 2700h Format: U16
	47:32	<b>Inverse Pixel Value 39</b>
		Default Value: 2700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 39</b>
		Default Value: 2700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 39</b>
		Default Value: 2700h Format: U16
158..159	63:48	<b>Forward R-ch Gamma Corrected Value 39</b>
		Default Value: 2700h Format: U16
	47:32	<b>Forward Pixel Value 39</b>
		Default Value: 2700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 39</b>
		Default Value: 2700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 39</b>
		Default Value: 2700h Format: U16
160..161	63:48	<b>Inverse R-ch Gamma Corrected Value 40</b>
		Default Value: 2800h Format: U16
	47:32	<b>Inverse Pixel Value 40</b>
		Default Value: 2800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 40</b>
		Default Value: 2800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 40</b>
		Default Value: 2800h Format: U16
162..163	63:48	<b>Forward R-ch Gamma Corrected Value 40</b>
		Default Value: 2800h Format: U16
	47:32	<b>Forward Pixel Value 40</b>
		Default Value: 2800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 40</b>
		Default Value: 2800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 40</b>
		Default Value: 2800h Format: U16
164..165	63:48	<b>Inverse R-ch Gamma Corrected Value 41</b>
		Default Value: 2900h Format: U16
	47:32	<b>Inverse Pixel Value 41</b>
		Default Value: 2900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 41</b>
		Default Value: 2900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 41</b>
		Default Value: 2900h Format: U16
166..167	63:48	<b>Forward R-ch Gamma Corrected Value 41</b>
		Default Value: 2900h Format: U16
	47:32	<b>Forward Pixel Value 41</b>
		Default Value: 2900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 41</b>
		Default Value: 2900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 41</b>
		Default Value: 2900h Format: U16
168..169	63:48	<b>Inverse R-ch Gamma Corrected Value 42</b>
		Default Value: 2a00h Format: U16
	47:32	<b>Inverse Pixel Value 42</b>
		Default Value: 2a00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 42</b>
		Default Value: 2a00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 42</b>
		Default Value: 2a00h Format: U16
170..171	63:48	<b>Forward R-ch Gamma Corrected Value 42</b>
		Default Value: 2a00h Format: U16
	47:32	<b>Forward Pixel Value 42</b>
		Default Value: 2a00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 42</b>
		Default Value: 2a00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 42</b>
		Default Value: 2a00h Format: U16
172..173	63:48	<b>Inverse R-ch Gamma Corrected Value 43</b>
		Default Value: 2b00h Format: U16
	47:32	<b>Inverse Pixel Value 43</b>
		Default Value: 2b00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 43</b>
		Default Value: 2b00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 43</b>
		Default Value: 2b00h Format: U16
174..175	63:48	<b>Forward R-ch Gamma Corrected Value 43</b>
		Default Value: 2b00h Format: U16
	47:32	<b>Forward Pixel Value 43</b>
		Default Value: 2b00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 43</b>
		Default Value: 2b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 43</b>
		Default Value: 2b00h Format: U16
176..177	63:48	<b>Inverse R-ch Gamma Corrected Value 44</b>
		Default Value: 2c00h Format: U16
	47:32	<b>Inverse Pixel Value 44</b>
		Default Value: 2c00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 44</b>
		Default Value: 2c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 44</b>
		Default Value: 2c00h Format: U16
178..179	63:48	<b>Forward R-ch Gamma Corrected Value 44</b>
		Default Value: 2c00h Format: U16
	47:32	<b>Forward Pixel Value 44</b>
		Default Value: 2c00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 44</b>
		Default Value: 2c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 44</b>
		Default Value: 2c00h Format: U16
180..181	63:48	<b>Inverse R-ch Gamma Corrected Value 45</b>
		Default Value: 2d00h Format: U16
	47:32	<b>Inverse Pixel Value 45</b>
		Default Value: 2d00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 45</b>
		Default Value: 2d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 45</b>
		Default Value: 2d00h Format: U16
182..183	63:48	<b>Forward R-ch Gamma Corrected Value 45</b>
		Default Value: 2d00h Format: U16
	47:32	<b>Forward Pixel Value 45</b>
		Default Value: 2d00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 45</b>
		Default Value: 2d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 45</b>
		Default Value: 2d00h Format: U16
184..185	63:48	<b>Inverse R-ch Gamma Corrected Value 46</b>
		Default Value: 2e00h Format: U16
	47:32	<b>Inverse Pixel Value 46</b>
		Default Value: 2e00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 46</b>
		Default Value: 2e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 46</b>
		Default Value: 2e00h Format: U16
186..187	63:48	<b>Forward R-ch Gamma Corrected Value 46</b>
		Default Value: 2e00h Format: U16
	47:32	<b>Forward Pixel Value 46</b>
		Default Value: 2e00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 46</b>
		Default Value: 2e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 46</b>
		Default Value: 2e00h Format: U16
188..189	63:48	<b>Inverse R-ch Gamma Corrected Value 47</b>
		Default Value: 2f00h Format: U16
	47:32	<b>Inverse Pixel Value 47</b>
		Default Value: 2f00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 47</b>
		Default Value: 2f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 47</b>
		Default Value: 2f00h Format: U16
190..191	63:48	<b>Forward R-ch Gamma Corrected Value 47</b>
		Default Value: 2f00h Format: U16
	47:32	<b>Forward Pixel Value 47</b>
		Default Value: 2f00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 47</b>
		Default Value: 2f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 47</b>
		Default Value: 2f00h Format: U16
192..193	63:48	<b>Inverse R-ch Gamma Corrected Value 48</b>
		Default Value: 3000h Format: U16
	47:32	<b>Inverse Pixel Value 48</b>
		Default Value: 3000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 48</b>
		Default Value: 3000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 48</b>
		Default Value: 3000h Format: U16
194..195	63:48	<b>Forward R-ch Gamma Corrected Value 48</b>
		Default Value: 3000h Format: U16
	47:32	<b>Forward Pixel Value 48</b>
		Default Value: 3000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 48</b>
		Default Value: 3000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 48</b>
		Default Value: 3000h Format: U16
196..197	63:48	<b>Inverse R-ch Gamma Corrected Value 49</b>
		Default Value: 3100h Format: U16
	47:32	<b>Inverse Pixel Value 49</b>
		Default Value: 3100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 49</b>
		Default Value: 3100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 49</b>
		Default Value: 3100h Format: U16
198..199	63:48	<b>Forward R-ch Gamma Corrected Value 49</b>
		Default Value: 3100h Format: U16
	47:32	<b>Forward Pixel Value 49</b>
		Default Value: 3100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 49</b>
		Default Value: 3100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 49</b>
		Default Value: 3100h Format: U16
200..201	63:48	<b>Inverse R-ch Gamma Corrected Value 50</b>
		Default Value: 3200h Format: U16
	47:32	<b>Inverse Pixel Value 50</b>
		Default Value: 3200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 50</b>
		Default Value: 3200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 50</b>
		Default Value: 3200h Format: U16
202..203	63:48	<b>Forward R-ch Gamma Corrected Value 50</b>
		Default Value: 3200h Format: U16
	47:32	<b>Forward Pixel Value 50</b>
		Default Value: 3200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 50</b>
		Default Value: 3200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 50</b>
		Default Value: 3200h Format: U16
204..205	63:48	<b>Inverse R-ch Gamma Corrected Value 51</b>
		Default Value: 3300h Format: U16
	47:32	<b>Inverse Pixel Value 51</b>
		Default Value: 3300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 51</b>
		Default Value: 3300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 51</b>
		Default Value: 3300h Format: U16
206..207	63:48	<b>Forward R-ch Gamma Corrected Value 51</b>
		Default Value: 3300h Format: U16
	47:32	<b>Forward Pixel Value 51</b>
		Default Value: 3300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 51</b>
		Default Value: 3300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 51</b>
		Default Value: 3300h Format: U16
208..209	63:48	<b>Inverse R-ch Gamma Corrected Value 52</b>
		Default Value: 3400h Format: U16
	47:32	<b>Inverse Pixel Value 52</b>
		Default Value: 3400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 52</b>
		Default Value: 3400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 52</b>
		Default Value: 3400h Format: U16
210..211	63:48	<b>Forward R-ch Gamma Corrected Value 52</b>
		Default Value: 3400h Format: U16
	47:32	<b>Forward Pixel Value 52</b>
		Default Value: 3400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 52</b>
		Default Value: 3400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 52</b>
		Default Value: 3400h Format: U16
212..213	63:48	<b>Inverse R-ch Gamma Corrected Value 53</b>
		Default Value: 3500h Format: U16
	47:32	<b>Inverse Pixel Value 53</b>
		Default Value: 3500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 53</b>
		Default Value: 3500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 53</b>
		Default Value: 3500h Format: U16
214..215	63:48	<b>Forward R-ch Gamma Corrected Value 53</b>
		Default Value: 3500h Format: U16
	47:32	<b>Forward Pixel Value 53</b>
		Default Value: 3500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 53</b>
		Default Value: 3500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 53</b>
		Default Value: 3500h Format: U16
216..217	63:48	<b>Inverse R-ch Gamma Corrected Value 54</b>
		Default Value: 3600h Format: U16
	47:32	<b>Inverse Pixel Value 54</b>
		Default Value: 3600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 54</b>
		Default Value: 3600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 54</b>
		Default Value: 3600h Format: U16
218..219	63:48	<b>Forward R-ch Gamma Corrected Value 54</b>
		Default Value: 3600h Format: U16
	47:32	<b>Forward Pixel Value 54</b>
		Default Value: 3600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 54</b>
		Default Value: 3600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 54</b>
		Default Value: 3600h Format: U16
220..221	63:48	<b>Inverse R-ch Gamma Corrected Value 55</b>
		Default Value: 3700h Format: U16
	47:32	<b>Inverse Pixel Value 55</b>
		Default Value: 3700h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 55</b>
		Default Value: 3700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 55</b>
		Default Value: 3700h Format: U16
222..223	63:48	<b>Forward R-ch Gamma Corrected Value 55</b>
		Default Value: 3700h Format: U16
	47:32	<b>Forward Pixel Value 55</b>
		Default Value: 3700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 55</b>
		Default Value: 3700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 55</b>
		Default Value: 3700h Format: U16
224..225	63:48	<b>Inverse R-ch Gamma Corrected Value 56</b>
		Default Value: 3800h Format: U16
	47:32	<b>Inverse Pixel Value 56</b>
		Default Value: 3800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 56</b>
		Default Value: 3800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 56</b>
		Default Value: 3800h Format: U16
226..227	63:48	<b>Forward R-ch Gamma Corrected Value 56</b>
		Default Value: 3800h Format: U16
	47:32	<b>Forward Pixel Value 56</b>
		Default Value: 3800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 56</b>
		Default Value: 3800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 56</b>
		Default Value: 3800h Format: U16
228..229	63:48	<b>Inverse R-ch Gamma Corrected Value 57</b>
		Default Value: 3900h Format: U16
	47:32	<b>Inverse Pixel Value 57</b>
		Default Value: 3900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 57</b>
		Default Value: 3900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 57</b>
		Default Value: 3900h Format: U16
230..231	63:48	<b>Forward R-ch Gamma Corrected Value 57</b>
		Default Value: 3900h Format: U16
	47:32	<b>Forward Pixel Value 57</b>
		Default Value: 3900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 57</b>
		Default Value: 3900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 57</b>
		Default Value: 3900h Format: U16
232..233	63:48	<b>Inverse R-ch Gamma Corrected Value 58</b>
		Default Value: 3a00h Format: U16
	47:32	<b>Inverse Pixel Value 58</b>
		Default Value: 3a00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 58</b>
		Default Value: 3a00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 58</b>
		Default Value: 3a00h Format: U16
234..235	63:48	<b>Forward R-ch Gamma Corrected Value 58</b>
		Default Value: 3a00h Format: U16
	47:32	<b>Forward Pixel Value 58</b>
		Default Value: 3a00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 58</b>
		Default Value: 3a00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 58</b>
		Default Value: 3a00h Format: U16
236..237	63:48	<b>Inverse R-ch Gamma Corrected Value 59</b>
		Default Value: 3b00h Format: U16
	47:32	<b>Inverse Pixel Value 59</b>
		Default Value: 3b00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 59</b>
		Default Value: 3b00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 59</b>
		Default Value: 3b00h Format: U16
238..239	63:48	<b>Forward R-ch Gamma Corrected Value 59</b>
		Default Value: 3b00h Format: U16
	47:32	<b>Forward Pixel Value 59</b>
		Default Value: 3b00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 59</b>
		Default Value: 3b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 59</b>
		Default Value: 3b00h Format: U16
240..241	63:48	<b>Inverse R-ch Gamma Corrected Value 60</b>
		Default Value: 3c00h Format: U16
	47:32	<b>Inverse Pixel Value 60</b>
		Default Value: 3c00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 60</b>
		Default Value: 3c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 60</b>
		Default Value: 3c00h Format: U16
242..243	63:48	<b>Forward R-ch Gamma Corrected Value 60</b>
		Default Value: 3c00h Format: U16
	47:32	<b>Forward Pixel Value 60</b>
		Default Value: 3c00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 60</b>
		Default Value: 3c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 60</b>
		Default Value: 3c00h Format: U16
244..245	63:48	<b>Inverse R-ch Gamma Corrected Value 61</b>
		Default Value: 3d00h Format: U16
	47:32	<b>Inverse Pixel Value 61</b>
		Default Value: 3d00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 61</b>
		Default Value: 3d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 61</b>
		Default Value: 3d00h Format: U16
246..247	63:48	<b>Forward R-ch Gamma Corrected Value 61</b>
		Default Value: 3d00h Format: U16
	47:32	<b>Forward Pixel Value 61</b>
		Default Value: 3d00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 61</b>
		Default Value: 3d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 61</b>
		Default Value: 3d00h Format: U16
248..249	63:48	<b>Inverse R-ch Gamma Corrected Value 62</b>
		Default Value: 3e00h Format: U16
	47:32	<b>Inverse Pixel Value 62</b>
		Default Value: 3e00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 62</b>
		Default Value: 3e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 62</b>
		Default Value: 3e00h Format: U16
250..251	63:48	<b>Forward R-ch Gamma Corrected Value 62</b>
		Default Value: 3e00h Format: U16
	47:32	<b>Forward Pixel Value 62</b>
		Default Value: 3e00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 62</b>
		Default Value: 3e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 62</b>
		Default Value: 3e00h Format: U16
252..253	63:48	<b>Inverse R-ch Gamma Corrected Value 63</b>
		Default Value: 3f00h Format: U16
	47:32	<b>Inverse Pixel Value 63</b>
		Default Value: 3f00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 63</b>
		Default Value: 3f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 63</b>
		Default Value: 3f00h Format: U16
254..255	63:48	<b>Forward R-ch Gamma Corrected Value 63</b>
		Default Value: 3f00h Format: U16
	47:32	<b>Forward Pixel Value 63</b>
		Default Value: 3f00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 63</b>
		Default Value: 3f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 63</b>
		Default Value: 3f00h Format: U16
256..257	63:48	<b>Inverse R-ch Gamma Corrected Value 64</b>
		Default Value: 4000h Format: U16
	47:32	<b>Inverse Pixel Value 64</b>
		Default Value: 4000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 64</b>
		Default Value: 4000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 64</b>
		Default Value: 4000h Format: U16
258..259	63:48	<b>Forward R-ch Gamma Corrected Value 64</b>
		Default Value: 4000h Format: U16
	47:32	<b>Forward Pixel Value 64</b>
		Default Value: 4000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 64</b>
		Default Value: 4000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 64</b>
		Default Value: 4000h Format: U16
260..261	63:48	<b>Inverse R-ch Gamma Corrected Value 65</b>
		Default Value: 4100h Format: U16
	47:32	<b>Inverse Pixel Value 65</b>
		Default Value: 4100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 65</b>
		Default Value: 4100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 65</b>
		Default Value: 4100h Format: U16
262..263	63:48	<b>Forward R-ch Gamma Corrected Value 65</b>
		Default Value: 4100h Format: U16
	47:32	<b>Forward Pixel Value 65</b>
		Default Value: 4100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 65</b>
		Default Value: 4100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 65</b>
		Default Value: 4100h Format: U16
264..265	63:48	<b>Inverse R-ch Gamma Corrected Value 66</b>
		Default Value: 4200h Format: U16
	47:32	<b>Inverse Pixel Value 66</b>
		Default Value: 4200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 66</b>
		Default Value: 4200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 66</b>
		Default Value: 4200h Format: U16
266..267	63:48	<b>Forward R-ch Gamma Corrected Value 66</b>
		Default Value: 4200h Format: U16
	47:32	<b>Forward Pixel Value 66</b>
		Default Value: 4200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 66</b>
		Default Value: 4200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 66</b>
		Default Value: 4200h Format: U16
268..269	63:48	<b>Inverse R-ch Gamma Corrected Value 67</b>
		Default Value: 4300h Format: U16
	47:32	<b>Inverse Pixel Value 67</b>
		Default Value: 4300h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 67</b>
		Default Value: 4300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 67</b>
		Default Value: 4300h Format: U16
270..271	63:48	<b>Forward R-ch Gamma Corrected Value 67</b>
		Default Value: 4300h Format: U16
	47:32	<b>Forward Pixel Value 67</b>
		Default Value: 4300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 67</b>
		Default Value: 4300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 67</b>
		Default Value: 4300h Format: U16
272..273	63:48	<b>Inverse R-ch Gamma Corrected Value 68</b>
		Default Value: 4400h Format: U16
	47:32	<b>Inverse Pixel Value 68</b>
		Default Value: 4400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 68</b>
		Default Value: 4400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 68</b>
		Default Value: 4400h Format: U16
274..275	63:48	<b>Forward R-ch Gamma Corrected Value 68</b>
		Default Value: 4400h Format: U16
	47:32	<b>Forward Pixel Value 68</b>
		Default Value: 4400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 68</b>
		Default Value: 4400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 68</b>
		Default Value: 4400h Format: U16
276..277	63:48	<b>Inverse R-ch Gamma Corrected Value 69</b>
		Default Value: 4500h Format: U16
	47:32	<b>Inverse Pixel Value 69</b>
		Default Value: 4500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 69</b>
		Default Value: 4500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 69</b>
		Default Value: 4500h Format: U16
278..279	63:48	<b>Forward R-ch Gamma Corrected Value 69</b>
		Default Value: 4500h Format: U16
	47:32	<b>Forward Pixel Value 69</b>
		Default Value: 4500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 69</b>
		Default Value: 4500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 69</b>
		Default Value: 4500h Format: U16
280..281	63:48	<b>Inverse R-ch Gamma Corrected Value 70</b>
		Default Value: 4600h Format: U16
	47:32	<b>Inverse Pixel Value 70</b>
		Default Value: 4600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 70</b>
		Default Value: 4600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 70</b>
		Default Value: 4600h Format: U16
282..283	63:48	<b>Forward R-ch Gamma Corrected Value 70</b>
		Default Value: 4600h Format: U16
	47:32	<b>Forward Pixel Value 70</b>
		Default Value: 4600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 70</b>
		Default Value: 4600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 70</b>
		Default Value: 4600h Format: U16
284..285	63:48	<b>Inverse R-ch Gamma Corrected Value 71</b>
		Default Value: 4700h Format: U16
	47:32	<b>Inverse Pixel Value 71</b>
		Default Value: 4700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 71</b>
		Default Value: 4700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 71</b>
		Default Value: 4700h Format: U16
286..287	63:48	<b>Forward R-ch Gamma Corrected Value 71</b>
		Default Value: 4700h Format: U16
	47:32	<b>Forward Pixel Value 71</b>
		Default Value: 4700h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 71</b>
		Default Value: 4700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 71</b>
		Default Value: 4700h Format: U16
288..289	63:48	<b>Inverse R-ch Gamma Corrected Value 72</b>
		Default Value: 4800h Format: U16
	47:32	<b>Inverse Pixel Value 72</b>
		Default Value: 4800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 72</b>
		Default Value: 4800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 72</b>
		Default Value: 4800h Format: U16
290..291	63:48	<b>Forward R-ch Gamma Corrected Value 72</b>
		Default Value: 4800h Format: U16
	47:32	<b>Forward Pixel Value 72</b>
		Default Value: 4800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 72</b>
		Default Value: 4800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 72</b>
		Default Value: 4800h Format: U16
292..293	63:48	<b>Inverse R-ch Gamma Corrected Value 73</b>
		Default Value: 4900h Format: U16
	47:32	<b>Inverse Pixel Value 73</b>
		Default Value: 4900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 73</b>
		Default Value: 4900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 73</b>
		Default Value: 4900h Format: U16
294..295	63:48	<b>Forward R-ch Gamma Corrected Value 73</b>
		Default Value: 4900h Format: U16
	47:32	<b>Forward Pixel Value 73</b>
		Default Value: 4900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 73</b>
		Default Value: 4900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 73</b>
		Default Value: 4900h Format: U16
296..297	63:48	<b>Inverse R-ch Gamma Corrected Value 74</b>
		Default Value: 4a00h Format: U16
	47:32	<b>Inverse Pixel Value 74</b>
		Default Value: 4a00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 74</b>
		Default Value: 4a00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 74</b>
		Default Value: 4a00h Format: U16
298..299	63:48	<b>Forward R-ch Gamma Corrected Value 74</b>
		Default Value: 4a00h Format: U16
	47:32	<b>Forward Pixel Value 74</b>
		Default Value: 4a00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 74</b>
		Default Value: 4a00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 74</b>
		Default Value: 4a00h Format: U16
300..301	63:48	<b>Inverse R-ch Gamma Corrected Value 75</b>
		Default Value: 4b00h Format: U16
	47:32	<b>Inverse Pixel Value 75</b>
		Default Value: 4b00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 75</b>
		Default Value: 4b00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 75</b>
		Default Value: 4b00h Format: U16
302..303	63:48	<b>Forward R-ch Gamma Corrected Value 75</b>
		Default Value: 4b00h Format: U16
	47:32	<b>Forward Pixel Value 75</b>
		Default Value: 4b00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 75</b>
		Default Value: 4b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 75</b>
		Default Value: 4b00h Format: U16
304..305	63:48	<b>Inverse R-ch Gamma Corrected Value 76</b>
		Default Value: 4c00h Format: U16
	47:32	<b>Inverse Pixel Value 76</b>
		Default Value: 4c00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 76</b>
		Default Value: 4c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 76</b>
		Default Value: 4c00h Format: U16
306..307	63:48	<b>Forward R-ch Gamma Corrected Value 76</b>
		Default Value: 4c00h Format: U16
	47:32	<b>Forward Pixel Value 76</b>
		Default Value: 4c00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 76</b>
		Default Value: 4c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 76</b>
		Default Value: 4c00h Format: U16
308..309	63:48	<b>Inverse R-ch Gamma Corrected Value 77</b>
		Default Value: 4d00h Format: U16
	47:32	<b>Inverse Pixel Value 77</b>
		Default Value: 4d00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 77</b>
		Default Value: 4d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 77</b>
		Default Value: 4d00h Format: U16
310..311	63:48	<b>Forward R-ch Gamma Corrected Value 77</b>
		Default Value: 4d00h Format: U16
	47:32	<b>Forward Pixel Value 77</b>
		Default Value: 4d00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 77</b>
		Default Value: 4d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 77</b>
		Default Value: 4d00h Format: U16
312..313	63:48	<b>Inverse R-ch Gamma Corrected Value 78</b>
		Default Value: 4e00h Format: U16
	47:32	<b>Inverse Pixel Value 78</b>
		Default Value: 4e00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 78</b>
		Default Value: 4e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 78</b>
		Default Value: 4e00h Format: U16
314..315	63:48	<b>Forward R-ch Gamma Corrected Value 78</b>
		Default Value: 4e00h Format: U16
	47:32	<b>Forward Pixel Value 78</b>
		Default Value: 4e00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 78</b>
		Default Value: 4e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 78</b>
		Default Value: 4e00h Format: U16
316..317	63:48	<b>Inverse R-ch Gamma Corrected Value 79</b>
		Default Value: 4f00h Format: U16
	47:32	<b>Inverse Pixel Value 79</b>
		Default Value: 4f00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 79</b>
		Default Value: 4f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 79</b>
		Default Value: 4f00h Format: U16
318..319	63:48	<b>Forward R-ch Gamma Corrected Value 79</b>
		Default Value: 4f00h Format: U16
	47:32	<b>Forward Pixel Value 79</b>
		Default Value: 4f00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 79</b>
		Default Value: 4f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 79</b>
		Default Value: 4f00h Format: U16
320..321	63:48	<b>Inverse R-ch Gamma Corrected Value 80</b>
		Default Value: 5000h Format: U16
	47:32	<b>Inverse Pixel Value 80</b>
		Default Value: 5000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 80</b>
		Default Value: 5000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 80</b>
		Default Value: 5000h Format: U16
322..323	63:48	<b>Forward R-ch Gamma Corrected Value 80</b>
		Default Value: 5000h Format: U16
	47:32	<b>Forward Pixel Value 80</b>
		Default Value: 5000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 80</b>
		Default Value: 5000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 80</b>
		Default Value: 5000h Format: U16
324..325	63:48	<b>Inverse R-ch Gamma Corrected Value 81</b>
		Default Value: 5100h Format: U16
	47:32	<b>Inverse Pixel Value 81</b>
		Default Value: 5100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 81</b>
		Default Value: 5100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 81</b>
		Default Value: 5100h Format: U16
326..327	63:48	<b>Forward R-ch Gamma Corrected Value 81</b>
		Default Value: 5100h Format: U16
	47:32	<b>Forward Pixel Value 81</b>
		Default Value: 5100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 81</b>
		Default Value: 5100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 81</b>
		Default Value: 5100h Format: U16
328..329	63:48	<b>Inverse R-ch Gamma Corrected Value 82</b>
		Default Value: 5200h Format: U16
	47:32	<b>Inverse Pixel Value 82</b>
		Default Value: 5200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 82</b>
		Default Value: 5200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 82</b>
		Default Value: 5200h Format: U16
330..331	63:48	<b>Forward R-ch Gamma Corrected Value 82</b>
		Default Value: 5200h Format: U16
	47:32	<b>Forward Pixel Value 82</b>
		Default Value: 5200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 82</b>
		Default Value: 5200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 82</b>
		Default Value: 5200h Format: U16
332..333	63:48	<b>Inverse R-ch Gamma Corrected Value 83</b>
		Default Value: 5300h Format: U16
	47:32	<b>Inverse Pixel Value 83</b>
		Default Value: 5300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 83</b>
		Default Value: 5300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 83</b>
		Default Value: 5300h Format: U16
334..335	63:48	<b>Forward R-ch Gamma Corrected Value 83</b>
		Default Value: 5300h Format: U16
	47:32	<b>Forward Pixel Value 83</b>
		Default Value: 5300h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 83</b>
		Default Value: 5300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 83</b>
		Default Value: 5300h Format: U16
336..337	63:48	<b>Inverse R-ch Gamma Corrected Value 84</b>
		Default Value: 5400h Format: U16
	47:32	<b>Inverse Pixel Value 84</b>
		Default Value: 5400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 84</b>
		Default Value: 5400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 84</b>
		Default Value: 5400h Format: U16
338..339	63:48	<b>Forward R-ch Gamma Corrected Value 84</b>
		Default Value: 5400h Format: U16
	47:32	<b>Forward Pixel Value 84</b>
		Default Value: 5400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 84</b>
		Default Value: 5400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 84</b>
		Default Value: 5400h Format: U16
340..341	63:48	<b>Inverse R-ch Gamma Corrected Value 85</b>
		Default Value: 5500h Format: U16
	47:32	<b>Inverse Pixel Value 85</b>
		Default Value: 5500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 85</b>
		Default Value: 5500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 85</b>
		Default Value: 5500h Format: U16
342..343	63:48	<b>Forward R-ch Gamma Corrected Value 85</b>
		Default Value: 5500h Format: U16
	47:32	<b>Forward Pixel Value 85</b>
		Default Value: 5500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 85</b>
		Default Value: 5500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 85</b>
		Default Value: 5500h Format: U16
344..345	63:48	<b>Inverse R-ch Gamma Corrected Value 86</b>
		Default Value: 5600h Format: U16
	47:32	<b>Inverse Pixel Value 86</b>
		Default Value: 5600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 86</b>
		Default Value: 5600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 86</b>
		Default Value: 5600h Format: U16
346..347	63:48	<b>Forward R-ch Gamma Corrected Value 86</b>
		Default Value: 5600h Format: U16
	47:32	<b>Forward Pixel Value 86</b>
		Default Value: 5600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 86</b>
		Default Value: 5600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 86</b>
		Default Value: 5600h Format: U16
348..349	63:48	<b>Inverse R-ch Gamma Corrected Value 87</b>
		Default Value: 5700h Format: U16
	47:32	<b>Inverse Pixel Value 87</b>
		Default Value: 5700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 87</b>
		Default Value: 5700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 87</b>
		Default Value: 5700h Format: U16
350..351	63:48	<b>Forward R-ch Gamma Corrected Value 87</b>
		Default Value: 5700h Format: U16
	47:32	<b>Forward Pixel Value 87</b>
		Default Value: 5700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 87</b>
		Default Value: 5700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 87</b>
		Default Value: 5700h Format: U16
352..353	63:48	<b>Inverse R-ch Gamma Corrected Value 88</b>
		Default Value: 5800h Format: U16
	47:32	<b>Inverse Pixel Value 88</b>
		Default Value: 5800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 88</b>
		Default Value: 5800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 88</b>
		Default Value: 5800h Format: U16
354..355	63:48	<b>Forward R-ch Gamma Corrected Value 88</b>
		Default Value: 5800h Format: U16
	47:32	<b>Forward Pixel Value 88</b>
		Default Value: 5800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 88</b>
		Default Value: 5800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 88</b>
		Default Value: 5800h Format: U16
356..357	63:48	<b>Inverse R-ch Gamma Corrected Value 89</b>
		Default Value: 5900h Format: U16
	47:32	<b>Inverse Pixel Value 89</b>
		Default Value: 5900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 89</b>
		Default Value: 5900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 89</b>
		Default Value: 5900h Format: U16
358..359	63:48	<b>Forward R-ch Gamma Corrected Value 89</b>
		Default Value: 5900h Format: U16
	47:32	<b>Forward Pixel Value 89</b>
		Default Value: 5900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 89</b>
		Default Value: 5900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 89</b>
		Default Value: 5900h Format: U16
360..361	63:48	<b>Inverse R-ch Gamma Corrected Value 90</b>
		Default Value: 5a00h Format: U16
	47:32	<b>Inverse Pixel Value 90</b>
		Default Value: 5a00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 90</b>
		Default Value: 5a00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 90</b>
		Default Value: 5a00h Format: U16
362..363	63:48	<b>Forward R-ch Gamma Corrected Value 90</b>
		Default Value: 5a00h Format: U16
	47:32	<b>Forward Pixel Value 90</b>
		Default Value: 5a00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 90</b>
		Default Value: 5a00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 90</b>
		Default Value: 5a00h Format: U16
364..365	63:48	<b>Inverse R-ch Gamma Corrected Value 91</b>
		Default Value: 5b00h Format: U16
	47:32	<b>Inverse Pixel Value 91</b>
		Default Value: 5b00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 91</b>
		Default Value: 5b00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 91</b>
		Default Value: 5b00h Format: U16
366..367	63:48	<b>Forward R-ch Gamma Corrected Value 91</b>
		Default Value: 5b00h Format: U16
	47:32	<b>Forward Pixel Value 91</b>
		Default Value: 5b00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 91</b>
		Default Value: 5b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 91</b>
		Default Value: 5b00h Format: U16
368..369	63:48	<b>Inverse R-ch Gamma Corrected Value 92</b>
		Default Value: 5c00h Format: U16
	47:32	<b>Inverse Pixel Value 92</b>
		Default Value: 5c00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 92</b>
		Default Value: 5c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 92</b>
		Default Value: 5c00h Format: U16
370..371	63:48	<b>Forward R-ch Gamma Corrected Value 92</b>
		Default Value: 5c00h Format: U16
	47:32	<b>Forward Pixel Value 92</b>
		Default Value: 5c00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 92</b>
		Default Value: 5c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 92</b>
		Default Value: 5c00h Format: U16
372..373	63:48	<b>Inverse R-ch Gamma Corrected Value 93</b>
		Default Value: 5d00h Format: U16
	47:32	<b>Inverse Pixel Value 93</b>
		Default Value: 5d00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 93</b>
		Default Value: 5d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 93</b>
		Default Value: 5d00h Format: U16
374..375	63:48	<b>Forward R-ch Gamma Corrected Value 93</b>
		Default Value: 5d00h Format: U16
	47:32	<b>Forward Pixel Value 93</b>
		Default Value: 5d00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 93</b>
		Default Value: 5d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 93</b>
		Default Value: 5d00h Format: U16
376..377	63:48	<b>Inverse R-ch Gamma Corrected Value 94</b>
		Default Value: 5e00h Format: U16
	47:32	<b>Inverse Pixel Value 94</b>
		Default Value: 5e00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 94</b>
		Default Value: 5e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 94</b>
		Default Value: 5e00h Format: U16
378..379	63:48	<b>Forward R-ch Gamma Corrected Value 94</b>
		Default Value: 5e00h Format: U16
	47:32	<b>Forward Pixel Value 94</b>
		Default Value: 5e00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 94</b>
		Default Value: 5e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 94</b>
		Default Value: 5e00h Format: U16
380..381	63:48	<b>Inverse R-ch Gamma Corrected Value 95</b>
		Default Value: 5f00h Format: U16
	47:32	<b>Inverse Pixel Value 95</b>
		Default Value: 5f00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 95</b>
		Default Value: 5f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 95</b>
		Default Value: 5f00h Format: U16
382..383	63:48	<b>Forward R-ch Gamma Corrected Value 95</b>
		Default Value: 5f00h Format: U16
	47:32	<b>Forward Pixel Value 95</b>
		Default Value: 5f00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 95</b>
		Default Value: 5f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 95</b>
		Default Value: 5f00h Format: U16
384..385	63:48	<b>Inverse R-ch Gamma Corrected Value 96</b>
		Default Value: 6000h Format: U16
	47:32	<b>Inverse Pixel Value 96</b>
		Default Value: 6000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 96</b>
		Default Value: 6000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 96</b>
		Default Value: 6000h Format: U16
386..387	63:48	<b>Forward R-ch Gamma Corrected Value 96</b>
		Default Value: 6000h Format: U16
	47:32	<b>Forward Pixel Value 96</b>
		Default Value: 6000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 96</b>
		Default Value: 6000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 96</b>
		Default Value: 6000h Format: U16
388..389	63:48	<b>Inverse R-ch Gamma Corrected Value 97</b>
		Default Value: 6100h Format: U16
	47:32	<b>Inverse Pixel Value 97</b>
		Default Value: 6100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 97</b>
		Default Value: 6100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 97</b>
		Default Value: 6100h Format: U16
390..391	63:48	<b>Forward R-ch Gamma Corrected Value 97</b>
		Default Value: 6100h Format: U16
	47:32	<b>Forward Pixel Value 97</b>
		Default Value: 6100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 97</b>
		Default Value: 6100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 97</b>
		Default Value: 6100h Format: U16
392..393	63:48	<b>Inverse R-ch Gamma Corrected Value 98</b>
		Default Value: 6200h Format: U16
	47:32	<b>Inverse Pixel Value 98</b>
		Default Value: 6200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 98</b>
		Default Value: 6200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 98</b>
		Default Value: 6200h Format: U16
394..395	63:48	<b>Forward R-ch Gamma Corrected Value 98</b>
		Default Value: 6200h Format: U16
	47:32	<b>Forward Pixel Value 98</b>
		Default Value: 6200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 98</b>
		Default Value: 6200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 98</b>
		Default Value: 6200h Format: U16
396..397	63:48	<b>Inverse R-ch Gamma Corrected Value 99</b>
		Default Value: 6300h Format: U16
	47:32	<b>Inverse Pixel Value 99</b>
		Default Value: 6300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 99</b>
		Default Value: 6300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 99</b>
		Default Value: 6300h Format: U16
398..399	63:48	<b>Forward R-ch Gamma Corrected Value 99</b>
		Default Value: 6300h Format: U16
	47:32	<b>Forward Pixel Value 99</b>
		Default Value: 6300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 99</b>
		Default Value: 6300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 99</b>
		Default Value: 6300h Format: U16
400..401	63:48	<b>Inverse R-ch Gamma Corrected Value 100</b>
		Default Value: 6400h Format: U16
	47:32	<b>Inverse Pixel Value 100</b>
		Default Value: 6400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 100</b>
		Default Value: 6400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 100</b>
		Default Value: 6400h Format: U16
402..403	63:48	<b>Forward R-ch Gamma Corrected Value 100</b>
		Default Value: 6400h Format: U16
	47:32	<b>Forward Pixel Value 100</b>
		Default Value: 6400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 100</b>
		Default Value: 6400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 100</b>
		Default Value: 6400h Format: U16
404..405	63:48	<b>Inverse R-ch Gamma Corrected Value 101</b>
		Default Value: 6500h Format: U16
	47:32	<b>Inverse Pixel Value 101</b>
		Default Value: 6500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 101</b>
		Default Value: 6500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 101</b>
		Default Value: 6500h Format: U16
406..407	63:48	<b>Forward R-ch Gamma Corrected Value 101</b>
		Default Value: 6500h Format: U16
	47:32	<b>Forward Pixel Value 101</b>
		Default Value: 6500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 101</b>
		Default Value: 6500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 101</b>
		Default Value: 6500h Format: U16
408..409	63:48	<b>Inverse R-ch Gamma Corrected Value 102</b>
		Default Value: 6600h Format: U16
	47:32	<b>Inverse Pixel Value 102</b>
		Default Value: 6600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 102</b>
		Default Value: 6600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 102</b>
		Default Value: 6600h Format: U16
410..411	63:48	<b>Forward R-ch Gamma Corrected Value 102</b>
		Default Value: 6600h Format: U16
	47:32	<b>Forward Pixel Value 102</b>
		Default Value: 6600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 102</b>
		Default Value: 6600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 102</b>
		Default Value: 6600h Format: U16
412..413	63:48	<b>Inverse R-ch Gamma Corrected Value 103</b>
		Default Value: 6700h Format: U16
	47:32	<b>Inverse Pixel Value 103</b>
		Default Value: 6700h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 103</b>
		Default Value: 6700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 103</b>
		Default Value: 6700h Format: U16
414..415	63:48	<b>Forward R-ch Gamma Corrected Value 103</b>
		Default Value: 6700h Format: U16
	47:32	<b>Forward Pixel Value 103</b>
		Default Value: 6700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 103</b>
		Default Value: 6700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 103</b>
		Default Value: 6700h Format: U16
416..417	63:48	<b>Inverse R-ch Gamma Corrected Value 104</b>
		Default Value: 6800h Format: U16
	47:32	<b>Inverse Pixel Value 104</b>
		Default Value: 6800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 104</b>
		Default Value: 6800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 104</b>
		Default Value: 6800h Format: U16
418..419	63:48	<b>Forward R-ch Gamma Corrected Value 104</b>
		Default Value: 6800h Format: U16
	47:32	<b>Forward Pixel Value 104</b>
		Default Value: 6800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 104</b>
		Default Value: 6800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 104</b>
		Default Value: 6800h Format: U16
420..421	63:48	<b>Inverse R-ch Gamma Corrected Value 105</b>
		Default Value: 6900h Format: U16
	47:32	<b>Inverse Pixel Value 105</b>
		Default Value: 6900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 105</b>
		Default Value: 6900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 105</b>
		Default Value: 6900h Format: U16
422..423	63:48	<b>Forward R-ch Gamma Corrected Value 105</b>
		Default Value: 6900h Format: U16
	47:32	<b>Forward Pixel Value 105</b>
		Default Value: 6900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 105</b>
		Default Value: 6900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 105</b>
		Default Value: 6900h Format: U16
424..425	63:48	<b>Inverse R-ch Gamma Corrected Value 106</b>
		Default Value: 6a00h Format: U16
	47:32	<b>Inverse Pixel Value 106</b>
		Default Value: 6a00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 106</b>
		Default Value: 6a00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 106</b>
		Default Value: 6a00h Format: U16
426..427	63:48	<b>Forward R-ch Gamma Corrected Value 106</b>
		Default Value: 6a00h Format: U16
	47:32	<b>Forward Pixel Value 106</b>
		Default Value: 6a00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 106</b>
		Default Value: 6a00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 106</b>
		Default Value: 6a00h Format: U16
428..429	63:48	<b>Inverse R-ch Gamma Corrected Value 107</b>
		Default Value: 6b00h Format: U16
	47:32	<b>Inverse Pixel Value 107</b>
		Default Value: 6b00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 107</b>
		Default Value: 6b00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 107</b>
		Default Value: 6b00h Format: U16
430..431	63:48	<b>Forward R-ch Gamma Corrected Value 107</b>
		Default Value: 6b00h Format: U16
	47:32	<b>Forward Pixel Value 107</b>
		Default Value: 6b00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 107</b>
		Default Value: 6b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 107</b>
		Default Value: 6b00h Format: U16
432..433	63:48	<b>Inverse R-ch Gamma Corrected Value 108</b>
		Default Value: 6c00h Format: U16
	47:32	<b>Inverse Pixel Value 108</b>
		Default Value: 6c00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 108</b>
		Default Value: 6c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 108</b>
		Default Value: 6c00h Format: U16
434..435	63:48	<b>Forward R-ch Gamma Corrected Value 108</b>
		Default Value: 6c00h Format: U16
	47:32	<b>Forward Pixel Value 108</b>
		Default Value: 6c00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 108</b>
		Default Value: 6c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 108</b>
		Default Value: 6c00h Format: U16
436..437	63:48	<b>Inverse R-ch Gamma Corrected Value 109</b>
		Default Value: 6d00h Format: U16
	47:32	<b>Inverse Pixel Value 109</b>
		Default Value: 6d00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 109</b>
		Default Value: 6d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 109</b>
		Default Value: 6d00h Format: U16
438..439	63:48	<b>Forward R-ch Gamma Corrected Value 109</b>
		Default Value: 6d00h Format: U16
	47:32	<b>Forward Pixel Value 109</b>
		Default Value: 6d00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 109</b>
		Default Value: 6d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 109</b>
		Default Value: 6d00h Format: U16
440..441	63:48	<b>Inverse R-ch Gamma Corrected Value 110</b>
		Default Value: 6e00h Format: U16
	47:32	<b>Inverse Pixel Value 110</b>
		Default Value: 6e00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 110</b>
		Default Value: 6e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 110</b>
		Default Value: 6e00h Format: U16
442..443	63:48	<b>Forward R-ch Gamma Corrected Value 110</b>
		Default Value: 6e00h Format: U16
	47:32	<b>Forward Pixel Value 110</b>
		Default Value: 6e00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 110</b>
		Default Value: 6e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 110</b>
		Default Value: 6e00h Format: U16
444..445	63:48	<b>Inverse R-ch Gamma Corrected Value 111</b>
		Default Value: 6f00h Format: U16
	47:32	<b>Inverse Pixel Value 111</b>
		Default Value: 6f00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 111</b>
		Default Value: 6f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 111</b>
		Default Value: 6f00h Format: U16
446..447	63:48	<b>Forward R-ch Gamma Corrected Value 111</b>
		Default Value: 6f00h Format: U16
	47:32	<b>Forward Pixel Value 111</b>
		Default Value: 6f00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 111</b>
		Default Value: 6f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 111</b>
		Default Value: 6f00h Format: U16
448..449	63:48	<b>Inverse R-ch Gamma Corrected Value 112</b>
		Default Value: 7000h Format: U16
	47:32	<b>Inverse Pixel Value 112</b>
		Default Value: 7000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 112</b>
		Default Value: 7000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 112</b>
		Default Value: 7000h Format: U16
450..451	63:48	<b>Forward R-ch Gamma Corrected Value 112</b>
		Default Value: 7000h Format: U16
	47:32	<b>Forward Pixel Value 112</b>
		Default Value: 7000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 112</b>
		Default Value: 7000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 112</b>
		Default Value: 7000h Format: U16
452..453	63:48	<b>Inverse R-ch Gamma Corrected Value 113</b>
		Default Value: 7100h Format: U16
	47:32	<b>Inverse Pixel Value 113</b>
		Default Value: 7100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 113</b>
		Default Value: 7100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 113</b>
		Default Value: 7100h Format: U16
454..455	63:48	<b>Forward R-ch Gamma Corrected Value 113</b>
		Default Value: 7100h Format: U16
	47:32	<b>Forward Pixel Value 113</b>
		Default Value: 7100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 113</b>
		Default Value: 7100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 113</b>
		Default Value: 7100h Format: U16
456..457	63:48	<b>Inverse R-ch Gamma Corrected Value 114</b>
		Default Value: 7200h Format: U16
	47:32	<b>Inverse Pixel Value 114</b>
		Default Value: 7200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 114</b>
		Default Value: 7200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 114</b>
		Default Value: 7200h Format: U16
458..459	63:48	<b>Forward R-ch Gamma Corrected Value 114</b>
		Default Value: 7200h Format: U16
	47:32	<b>Forward Pixel Value 114</b>
		Default Value: 7200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 114</b>
		Default Value: 7200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 114</b>
		Default Value: 7200h Format: U16
460..461	63:48	<b>Inverse R-ch Gamma Corrected Value 115</b>
		Default Value: 7300h Format: U16
	47:32	<b>Inverse Pixel Value 115</b>
		Default Value: 7300h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 115</b>
		Default Value: 7300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 115</b>
		Default Value: 7300h Format: U16
462..463	63:48	<b>Forward R-ch Gamma Corrected Value 115</b>
		Default Value: 7300h Format: U16
	47:32	<b>Forward Pixel Value 115</b>
		Default Value: 7300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 115</b>
		Default Value: 7300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 115</b>
		Default Value: 7300h Format: U16
464..465	63:48	<b>Inverse R-ch Gamma Corrected Value 116</b>
		Default Value: 7400h Format: U16
	47:32	<b>Inverse Pixel Value 116</b>
		Default Value: 7400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 116</b>
		Default Value: 7400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 116</b>
		Default Value: 7400h Format: U16
466..467	63:48	<b>Forward R-ch Gamma Corrected Value 116</b>
		Default Value: 7400h Format: U16
	47:32	<b>Forward Pixel Value 116</b>
		Default Value: 7400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 116</b>
		Default Value: 7400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 116</b>
		Default Value: 7400h Format: U16
468..469	63:48	<b>Inverse R-ch Gamma Corrected Value 117</b>
		Default Value: 7500h Format: U16
	47:32	<b>Inverse Pixel Value 117</b>
		Default Value: 7500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 117</b>
		Default Value: 7500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 117</b>
		Default Value: 7500h Format: U16
470..471	63:48	<b>Forward R-ch Gamma Corrected Value 117</b>
		Default Value: 7500h Format: U16
	47:32	<b>Forward Pixel Value 117</b>
		Default Value: 7500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 117</b>
		Default Value: 7500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 117</b>
		Default Value: 7500h Format: U16
472..473	63:48	<b>Inverse R-ch Gamma Corrected Value 118</b>
		Default Value: 7600h Format: U16
	47:32	<b>Inverse Pixel Value 118</b>
		Default Value: 7600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 118</b>
		Default Value: 7600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 118</b>
		Default Value: 7600h Format: U16
474..475	63:48	<b>Forward R-ch Gamma Corrected Value 118</b>
		Default Value: 7600h Format: U16
	47:32	<b>Forward Pixel Value 118</b>
		Default Value: 7600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 118</b>
		Default Value: 7600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 118</b>
		Default Value: 7600h Format: U16
476..477	63:48	<b>Inverse R-ch Gamma Corrected Value 119</b>
		Default Value: 7700h Format: U16
	47:32	<b>Inverse Pixel Value 119</b>
		Default Value: 7700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 119</b>
		Default Value: 7700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 119</b>
		Default Value: 7700h Format: U16
478..479	63:48	<b>Forward R-ch Gamma Corrected Value 119</b>
		Default Value: 7700h Format: U16
	47:32	<b>Forward Pixel Value 119</b>
		Default Value: 7700h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 119</b>
		Default Value: 7700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 119</b>
		Default Value: 7700h Format: U16
480..481	63:48	<b>Inverse R-ch Gamma Corrected Value 120</b>
		Default Value: 7800h Format: U16
	47:32	<b>Inverse Pixel Value 120</b>
		Default Value: 7800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 120</b>
		Default Value: 7800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 120</b>
		Default Value: 7800h Format: U16
482..483	63:48	<b>Forward R-ch Gamma Corrected Value 120</b>
		Default Value: 7800h Format: U16
	47:32	<b>Forward Pixel Value 120</b>
		Default Value: 7800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 120</b>
		Default Value: 7800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 120</b>
		Default Value: 7800h Format: U16
484..485	63:48	<b>Inverse R-ch Gamma Corrected Value 121</b>
		Default Value: 7900h Format: U16
	47:32	<b>Inverse Pixel Value 121</b>
		Default Value: 7900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 121</b>
		Default Value: 7900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 121</b>
		Default Value: 7900h Format: U16
486..487	63:48	<b>Forward R-ch Gamma Corrected Value 121</b>
		Default Value: 7900h Format: U16
	47:32	<b>Forward Pixel Value 121</b>
		Default Value: 7900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 121</b>
		Default Value: 7900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 121</b>
		Default Value: 7900h Format: U16
488..489	63:48	<b>Inverse R-ch Gamma Corrected Value 122</b>
		Default Value: 7a00h Format: U16
	47:32	<b>Inverse Pixel Value 122</b>
		Default Value: 7a00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 122</b>
		Default Value: 7a00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 122</b>
		Default Value: 7a00h Format: U16
490..491	63:48	<b>Forward R-ch Gamma Corrected Value 122</b>
		Default Value: 7a00h Format: U16
	47:32	<b>Forward Pixel Value 122</b>
		Default Value: 7a00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 122</b>
		Default Value: 7a00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 122</b>
		Default Value: 7a00h Format: U16
492..493	63:48	<b>Inverse R-ch Gamma Corrected Value 123</b>
		Default Value: 7b00h Format: U16
	47:32	<b>Inverse Pixel Value 123</b>
		Default Value: 7b00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 123</b>
		Default Value: 7b00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 123</b>
		Default Value: 7b00h Format: U16
494..495	63:48	<b>Forward R-ch Gamma Corrected Value 123</b>
		Default Value: 7b00h Format: U16
	47:32	<b>Forward Pixel Value 123</b>
		Default Value: 7b00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 123</b>
		Default Value: 7b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 123</b>
		Default Value: 7b00h Format: U16
496..497	63:48	<b>Inverse R-ch Gamma Corrected Value 124</b>
		Default Value: 7c00h Format: U16
	47:32	<b>Inverse Pixel Value 124</b>
		Default Value: 7c00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 124</b>
		Default Value: 7c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 124</b>
		Default Value: 7c00h Format: U16
498..499	63:48	<b>Forward R-ch Gamma Corrected Value 124</b>
		Default Value: 7c00h Format: U16
	47:32	<b>Forward Pixel Value 124</b>
		Default Value: 7c00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 124</b>
		Default Value: 7c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 124</b>
		Default Value: 7c00h Format: U16
500..501	63:48	<b>Inverse R-ch Gamma Corrected Value 125</b>
		Default Value: 7d00h Format: U16
	47:32	<b>Inverse Pixel Value 125</b>
		Default Value: 7d00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 125</b>
		Default Value: 7d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 125</b>
		Default Value: 7d00h Format: U16
502..503	63:48	<b>Forward R-ch Gamma Corrected Value 125</b>
		Default Value: 7d00h Format: U16
	47:32	<b>Forward Pixel Value 125</b>
		Default Value: 7d00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 125</b>
		Default Value: 7d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 125</b>
		Default Value: 7d00h Format: U16
504..505	63:48	<b>Inverse R-ch Gamma Corrected Value 126</b>
		Default Value: 7e00h Format: U16
	47:32	<b>Inverse Pixel Value 126</b>
		Default Value: 7e00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 126</b>
		Default Value: 7e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 126</b>
		Default Value: 7e00h Format: U16
506..507	63:48	<b>Forward R-ch Gamma Corrected Value 126</b>
		Default Value: 7e00h Format: U16
	47:32	<b>Forward Pixel Value 126</b>
		Default Value: 7e00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 126</b>
		Default Value: 7e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 126</b>
		Default Value: 7e00h Format: U16
508..509	63:48	<b>Inverse R-ch Gamma Corrected Value 127</b>
		Default Value: 7f00h Format: U16
	47:32	<b>Inverse Pixel Value 127</b>
		Default Value: 7f00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 127</b>
		Default Value: 7f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 127</b>
		Default Value: 7f00h Format: U16
510..511	63:48	<b>Forward R-ch Gamma Corrected Value 127</b>
		Default Value: 7f00h Format: U16
	47:32	<b>Forward Pixel Value 127</b>
		Default Value: 7f00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 127</b>
		Default Value: 7f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 127</b>
		Default Value: 7f00h Format: U16
512..513	63:48	<b>Inverse R-ch Gamma Corrected Value 128</b>
		Default Value: 8000h Format: U16
	47:32	<b>Inverse Pixel Value 128</b>
		Default Value: 8000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 128</b>
		Default Value: 8000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 128</b>
		Default Value: 8000h Format: U16
514..515	63:48	<b>Forward R-ch Gamma Corrected Value 128</b>
		Default Value: 8000h Format: U16
	47:32	<b>Forward Pixel Value 128</b>
		Default Value: 8000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 128</b>
		Default Value: 8000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 128</b>
		Default Value: 8000h Format: U16
516..517	63:48	<b>Inverse R-ch Gamma Corrected Value 129</b>
		Default Value: 8100h Format: U16
	47:32	<b>Inverse Pixel Value 129</b>
		Default Value: 8100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 129</b>
		Default Value: 8100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 129</b>
		Default Value: 8100h Format: U16
518..519	63:48	<b>Forward R-ch Gamma Corrected Value 129</b>
		Default Value: 8100h Format: U16
	47:32	<b>Forward Pixel Value 129</b>
		Default Value: 8100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 129</b>
		Default Value: 8100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 129</b>
		Default Value: 8100h Format: U16
520..521	63:48	<b>Inverse R-ch Gamma Corrected Value 130</b>
		Default Value: 8200h Format: U16
	47:32	<b>Inverse Pixel Value 130</b>
		Default Value: 8200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 130</b>
		Default Value: 8200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 130</b>
		Default Value: 8200h Format: U16
522..523	63:48	<b>Forward R-ch Gamma Corrected Value 130</b>
		Default Value: 8200h Format: U16
	47:32	<b>Forward Pixel Value 130</b>
		Default Value: 8200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 130</b>
		Default Value: 8200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 130</b>
		Default Value: 8200h Format: U16
524..525	63:48	<b>Inverse R-ch Gamma Corrected Value 131</b>
		Default Value: 8300h Format: U16
	47:32	<b>Inverse Pixel Value 131</b>
		Default Value: 8300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 131</b>
		Default Value: 8300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 131</b>
		Default Value: 8300h Format: U16
526..527	63:48	<b>Forward R-ch Gamma Corrected Value 131</b>
		Default Value: 8300h Format: U16
	47:32	<b>Forward Pixel Value 131</b>
		Default Value: 8300h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 131</b>
		Default Value: 8300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 131</b>
		Default Value: 8300h Format: U16
528..529	63:48	<b>Inverse R-ch Gamma Corrected Value 132</b>
		Default Value: 8400h Format: U16
	47:32	<b>Inverse Pixel Value 132</b>
		Default Value: 8400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 132</b>
		Default Value: 8400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 132</b>
		Default Value: 8400h Format: U16
530..531	63:48	<b>Forward R-ch Gamma Corrected Value 132</b>
		Default Value: 8400h Format: U16
	47:32	<b>Forward Pixel Value 132</b>
		Default Value: 8400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 132</b>
		Default Value: 8400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 132</b>
		Default Value: 8400h Format: U16
532..533	63:48	<b>Inverse R-ch Gamma Corrected Value 133</b>
		Default Value: 8500h Format: U16
	47:32	<b>Inverse Pixel Value 133</b>
		Default Value: 8500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 133</b>
		Default Value: 8500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 133</b>
		Default Value: 8500h Format: U16
534..535	63:48	<b>Forward R-ch Gamma Corrected Value 133</b>
		Default Value: 8500h Format: U16
	47:32	<b>Forward Pixel Value 133</b>
		Default Value: 8500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 133</b>
		Default Value: 8500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 133</b>
		Default Value: 8500h Format: U16
536..537	63:48	<b>Inverse R-ch Gamma Corrected Value 134</b>
		Default Value: 8600h Format: U16
	47:32	<b>Inverse Pixel Value 134</b>
		Default Value: 8600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 134</b>
		Default Value: 8600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 134</b>
		Default Value: 8600h Format: U16
538..539	63:48	<b>Forward R-ch Gamma Corrected Value 134</b>
		Default Value: 8600h Format: U16
	47:32	<b>Forward Pixel Value 134</b>
		Default Value: 8600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 134</b>
		Default Value: 8600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 134</b>
		Default Value: 8600h Format: U16
540..541	63:48	<b>Inverse R-ch Gamma Corrected Value 135</b>
		Default Value: 8700h Format: U16
	47:32	<b>Inverse Pixel Value 135</b>
		Default Value: 8700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 135</b>
		Default Value: 8700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 135</b>
		Default Value: 8700h Format: U16
542..543	63:48	<b>Forward R-ch Gamma Corrected Value 135</b>
		Default Value: 8700h Format: U16
	47:32	<b>Forward Pixel Value 135</b>
		Default Value: 8700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 135</b>
		Default Value: 8700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 135</b>
		Default Value: 8700h Format: U16
544..545	63:48	<b>Inverse R-ch Gamma Corrected Value 136</b>
		Default Value: 8800h Format: U16
	47:32	<b>Inverse Pixel Value 136</b>
		Default Value: 8800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 136</b>
		Default Value: 8800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 136</b>
		Default Value: 8800h Format: U16
546..547	63:48	<b>Forward R-ch Gamma Corrected Value 136</b>
		Default Value: 8800h Format: U16
	47:32	<b>Forward Pixel Value 136</b>
		Default Value: 8800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 136</b>
		Default Value: 8800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 136</b>
		Default Value: 8800h Format: U16
548..549	63:48	<b>Inverse R-ch Gamma Corrected Value 137</b>
		Default Value: 8900h Format: U16
	47:32	<b>Inverse Pixel Value 137</b>
		Default Value: 8900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 137</b>
		Default Value: 8900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 137</b>
		Default Value: 8900h Format: U16
550..551	63:48	<b>Forward R-ch Gamma Corrected Value 137</b>
		Default Value: 8900h Format: U16
	47:32	<b>Forward Pixel Value 137</b>
		Default Value: 8900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 137</b>
		Default Value: 8900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 137</b>
		Default Value: 8900h Format: U16
552..553	63:48	<b>Inverse R-ch Gamma Corrected Value 138</b>
		Default Value: 8a00h Format: U16
	47:32	<b>Inverse Pixel Value 138</b>
		Default Value: 8a00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 138</b>
		Default Value: 8a00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 138</b>
		Default Value: 8a00h Format: U16
554..555	63:48	<b>Forward R-ch Gamma Corrected Value 138</b>
		Default Value: 8a00h Format: U16
	47:32	<b>Forward Pixel Value 138</b>
		Default Value: 8a00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 138</b>
		Default Value: 8a00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 138</b>
		Default Value: 8a00h Format: U16
556..557	63:48	<b>Inverse R-ch Gamma Corrected Value 139</b>
		Default Value: 8b00h Format: U16
	47:32	<b>Inverse Pixel Value 139</b>
		Default Value: 8b00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 139</b>
		Default Value: 8b00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 139</b>
		Default Value: 8b00h Format: U16
558..559	63:48	<b>Forward R-ch Gamma Corrected Value 139</b>
		Default Value: 8b00h Format: U16
	47:32	<b>Forward Pixel Value 139</b>
		Default Value: 8b00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 139</b>
		Default Value: 8b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 139</b>
		Default Value: 8b00h Format: U16
560..561	63:48	<b>Inverse R-ch Gamma Corrected Value 140</b>
		Default Value: 8c00h Format: U16
	47:32	<b>Inverse Pixel Value 140</b>
		Default Value: 8c00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 140</b>
		Default Value: 8c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 140</b>
		Default Value: 8c00h Format: U16
562..563	63:48	<b>Forward R-ch Gamma Corrected Value 140</b>
		Default Value: 8c00h Format: U16
	47:32	<b>Forward Pixel Value 140</b>
		Default Value: 8c00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 140</b>
		Default Value: 8c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 140</b>
		Default Value: 8c00h Format: U16
564..565	63:48	<b>Inverse R-ch Gamma Corrected Value 141</b>
		Default Value: 8d00h Format: U16
	47:32	<b>Inverse Pixel Value 141</b>
		Default Value: 8d00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 141</b>
		Default Value: 8d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 141</b>
		Default Value: 8d00h Format: U16
566..567	63:48	<b>Forward R-ch Gamma Corrected Value 141</b>
		Default Value: 8d00h Format: U16
	47:32	<b>Forward Pixel Value 141</b>
		Default Value: 8d00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 141</b>
		Default Value: 8d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 141</b>
		Default Value: 8d00h Format: U16
568..569	63:48	<b>Inverse R-ch Gamma Corrected Value 142</b>
		Default Value: 8e00h Format: U16
	47:32	<b>Inverse Pixel Value 142</b>
		Default Value: 8e00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 142</b>
		Default Value: 8e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 142</b>
		Default Value: 8e00h Format: U16
570..571	63:48	<b>Forward R-ch Gamma Corrected Value 142</b>
		Default Value: 8e00h Format: U16
	47:32	<b>Forward Pixel Value 142</b>
		Default Value: 8e00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 142</b>
		Default Value: 8e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 142</b>
		Default Value: 8e00h Format: U16
572..573	63:48	<b>Inverse R-ch Gamma Corrected Value 143</b>
		Default Value: 8f00h Format: U16
	47:32	<b>Inverse Pixel Value 143</b>
		Default Value: 8f00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 143</b>
		Default Value: 8f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 143</b>
		Default Value: 8f00h Format: U16
574..575	63:48	<b>Forward R-ch Gamma Corrected Value 143</b>
		Default Value: 8f00h Format: U16
	47:32	<b>Forward Pixel Value 143</b>
		Default Value: 8f00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 143</b>
		Default Value: 8f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 143</b>
		Default Value: 8f00h Format: U16
576..577	63:48	<b>Inverse R-ch Gamma Corrected Value 144</b>
		Default Value: 9000h Format: U16
	47:32	<b>Inverse Pixel Value 144</b>
		Default Value: 9000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 144</b>
		Default Value: 9000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 144</b>
		Default Value: 9000h Format: U16
578..579	63:48	<b>Forward R-ch Gamma Corrected Value 144</b>
		Default Value: 9000h Format: U16
	47:32	<b>Forward Pixel Value 144</b>
		Default Value: 9000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 144</b>
		Default Value: 9000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 144</b>
		Default Value: 9000h Format: U16
580..581	63:48	<b>Inverse R-ch Gamma Corrected Value 145</b>
		Default Value: 9100h Format: U16
	47:32	<b>Inverse Pixel Value 145</b>
		Default Value: 9100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 145</b>
		Default Value: 9100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 145</b>
		Default Value: 9100h Format: U16
582..583	63:48	<b>Forward R-ch Gamma Corrected Value 145</b>
		Default Value: 9100h Format: U16
	47:32	<b>Forward Pixel Value 145</b>
		Default Value: 9100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 145</b>
		Default Value: 9100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 145</b>
		Default Value: 9100h Format: U16
584..585	63:48	<b>Inverse R-ch Gamma Corrected Value 146</b>
		Default Value: 9200h Format: U16
	47:32	<b>Inverse Pixel Value 146</b>
		Default Value: 9200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 146</b>
		Default Value: 9200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 146</b>
		Default Value: 9200h Format: U16
586..587	63:48	<b>Forward R-ch Gamma Corrected Value 146</b>
		Default Value: 9200h Format: U16
	47:32	<b>Forward Pixel Value 146</b>
		Default Value: 9200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 146</b>
		Default Value: 9200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 146</b>
		Default Value: 9200h Format: U16
588..589	63:48	<b>Inverse R-ch Gamma Corrected Value 147</b>
		Default Value: 9300h Format: U16
	47:32	<b>Inverse Pixel Value 147</b>
		Default Value: 9300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 147</b>
		Default Value: 9300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 147</b>
		Default Value: 9300h Format: U16
590..591	63:48	<b>Forward R-ch Gamma Corrected Value 147</b>
		Default Value: 9300h Format: U16
	47:32	<b>Forward Pixel Value 147</b>
		Default Value: 9300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 147</b>
		Default Value: 9300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 147</b>
		Default Value: 9300h Format: U16
592..593	63:48	<b>Inverse R-ch Gamma Corrected Value 148</b>
		Default Value: 9400h Format: U16
	47:32	<b>Inverse Pixel Value 148</b>
		Default Value: 9400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 148</b>
		Default Value: 9400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 148</b>
		Default Value: 9400h Format: U16
594..595	63:48	<b>Forward R-ch Gamma Corrected Value 148</b>
		Default Value: 9400h Format: U16
	47:32	<b>Forward Pixel Value 148</b>
		Default Value: 9400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 148</b>
		Default Value: 9400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 148</b>
		Default Value: 9400h Format: U16
596..597	63:48	<b>Inverse R-ch Gamma Corrected Value 149</b>
		Default Value: 9500h Format: U16
	47:32	<b>Inverse Pixel Value 149</b>
		Default Value: 9500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 149</b>
		Default Value: 9500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 149</b>
		Default Value: 9500h Format: U16
598..599	63:48	<b>Forward R-ch Gamma Corrected Value 149</b>
		Default Value: 9500h Format: U16
	47:32	<b>Forward Pixel Value 149</b>
		Default Value: 9500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 149</b>
		Default Value: 9500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 149</b>
		Default Value: 9500h Format: U16
600..601	63:48	<b>Inverse R-ch Gamma Corrected Value 150</b>
		Default Value: 9600h Format: U16
	47:32	<b>Inverse Pixel Value 150</b>
		Default Value: 9600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 150</b>
		Default Value: 9600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 150</b>
		Default Value: 9600h Format: U16
602..603	63:48	<b>Forward R-ch Gamma Corrected Value 150</b>
		Default Value: 9600h Format: U16
	47:32	<b>Forward Pixel Value 150</b>
		Default Value: 9600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 150</b>
		Default Value: 9600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 150</b>
		Default Value: 9600h Format: U16
604..605	63:48	<b>Inverse R-ch Gamma Corrected Value 151</b>
		Default Value: 9700h Format: U16
	47:32	<b>Inverse Pixel Value 151</b>
		Default Value: 9700h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 151</b>
		Default Value: 9700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 151</b>
		Default Value: 9700h Format: U16
606..607	63:48	<b>Forward R-ch Gamma Corrected Value 151</b>
		Default Value: 9700h Format: U16
	47:32	<b>Forward Pixel Value 151</b>
		Default Value: 9700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 151</b>
		Default Value: 9700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 151</b>
		Default Value: 9700h Format: U16
608..609	63:48	<b>Inverse R-ch Gamma Corrected Value 152</b>
		Default Value: 9800h Format: U16
	47:32	<b>Inverse Pixel Value 152</b>
		Default Value: 9800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 152</b>
		Default Value: 9800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 152</b>
		Default Value: 9800h Format: U16
610..611	63:48	<b>Forward R-ch Gamma Corrected Value 152</b>
		Default Value: 9800h Format: U16
	47:32	<b>Forward Pixel Value 152</b>
		Default Value: 9800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 152</b>
		Default Value: 9800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 152</b>
		Default Value: 9800h Format: U16
612..613	63:48	<b>Inverse R-ch Gamma Corrected Value 153</b>
		Default Value: 9900h Format: U16
	47:32	<b>Inverse Pixel Value 153</b>
		Default Value: 9900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 153</b>
		Default Value: 9900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 153</b>
		Default Value: 9900h Format: U16
614..615	63:48	<b>Forward R-ch Gamma Corrected Value 153</b>
		Default Value: 9900h Format: U16
	47:32	<b>Forward Pixel Value 153</b>
		Default Value: 9900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 153</b>
		Default Value: 9900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 153</b>
		Default Value: 9900h Format: U16
616..617	63:48	<b>Inverse R-ch Gamma Corrected Value 154</b>
		Default Value: 9a00h Format: U16
	47:32	<b>Inverse Pixel Value 154</b>
		Default Value: 9a00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>			
	31:16	<b>Inverse B-ch Gamma Corrected Value 154</b>	
		Default Value: 9a00h Format: U16	
	15:0	<b>Inverse G-ch Gamma Corrected Value 154</b>	
		Default Value: 9a00h Format: U16	
	618..619	63:48	<b>Forward R-ch Gamma Corrected Value 154</b>
			Default Value: 9a00h Format: U16
47:32		<b>Forward Pixel Value 154</b>	
		Default Value: 9a00h Format: U16	
31:16		<b>Forward B-ch Gamma Corrected Value 154</b>	
		Default Value: 9a00h Format: U16	
15:0		<b>Forward G-ch Gamma Corrected Value 154</b>	
		Default Value: 9a00h Format: U16	
620..621	63:48	<b>Inverse R-ch Gamma Corrected Value 155</b>	
		Default Value: 9b00h Format: U16	
	47:32	<b>Inverse Pixel Value 155</b>	
		Default Value: 9b00h Format: U16	
	31:16	<b>Inverse B-ch Gamma Corrected Value 155</b>	
		Default Value: 9b00h Format: U16	
	15:0	<b>Inverse G-ch Gamma Corrected Value 155</b>	
		Default Value: 9b00h Format: U16	
622..623	63:48	<b>Forward R-ch Gamma Corrected Value 155</b>	
		Default Value: 9b00h Format: U16	
	47:32	<b>Forward Pixel Value 155</b>	
		Default Value: 9b00h Format: U16	



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 155</b>
		Default Value: 9b00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 155</b>
		Default Value: 9b00h Format: U16
624..625	63:48	<b>Inverse R-ch Gamma Corrected Value 156</b>
		Default Value: 9c00h Format: U16
	47:32	<b>Inverse Pixel Value 156</b>
		Default Value: 9c00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 156</b>
		Default Value: 9c00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 156</b>
		Default Value: 9c00h Format: U16
626..627	63:48	<b>Forward R-ch Gamma Corrected Value 156</b>
		Default Value: 9c00h Format: U16
	47:32	<b>Forward Pixel Value 156</b>
		Default Value: 9c00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 156</b>
		Default Value: 9c00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 156</b>
		Default Value: 9c00h Format: U16
628..629	63:48	<b>Inverse R-ch Gamma Corrected Value 157</b>
		Default Value: 9d00h Format: U16
	47:32	<b>Inverse Pixel Value 157</b>
		Default Value: 9d00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 157</b>
		Default Value: 9d00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 157</b>
		Default Value: 9d00h Format: U16
630..631	63:48	<b>Forward R-ch Gamma Corrected Value 157</b>
		Default Value: 9d00h Format: U16
	47:32	<b>Forward Pixel Value 157</b>
		Default Value: 9d00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 157</b>
		Default Value: 9d00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 157</b>
		Default Value: 9d00h Format: U16
632..633	63:48	<b>Inverse R-ch Gamma Corrected Value 158</b>
		Default Value: 9e00h Format: U16
	47:32	<b>Inverse Pixel Value 158</b>
		Default Value: 9e00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 158</b>
		Default Value: 9e00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 158</b>
		Default Value: 9e00h Format: U16
634..635	63:48	<b>Forward R-ch Gamma Corrected Value 158</b>
		Default Value: 9e00h Format: U16
	47:32	<b>Forward Pixel Value 158</b>
		Default Value: 9e00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 158</b>
		Default Value: 9e00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 158</b>
		Default Value: 9e00h Format: U16
636..637	63:48	<b>Inverse R-ch Gamma Corrected Value 159</b>
		Default Value: 9f00h Format: U16
	47:32	<b>Inverse Pixel Value 159</b>
		Default Value: 9f00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 159</b>
		Default Value: 9f00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 159</b>
		Default Value: 9f00h Format: U16
638..639	63:48	<b>Forward R-ch Gamma Corrected Value 159</b>
		Default Value: 9f00h Format: U16
	47:32	<b>Forward Pixel Value 159</b>
		Default Value: 9f00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 159</b>
		Default Value: 9f00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 159</b>
		Default Value: 9f00h Format: U16
640..641	63:48	<b>Inverse R-ch Gamma Corrected Value 160</b>
		Default Value: a000h Format: U16
	47:32	<b>Inverse Pixel Value 160</b>
		Default Value: a000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 160</b>
		Default Value: a000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 160</b>
		Default Value: a000h Format: U16
642..643	63:48	<b>Forward R-ch Gamma Corrected Value 160</b>
		Default Value: a000h Format: U16
	47:32	<b>Forward Pixel Value 160</b>
		Default Value: a000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 160</b>
		Default Value: a000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 160</b>
		Default Value: a000h Format: U16
644..645	63:48	<b>Inverse R-ch Gamma Corrected Value 161</b>
		Default Value: a100h Format: U16
	47:32	<b>Inverse Pixel Value 161</b>
		Default Value: a100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 161</b>
		Default Value: a100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 161</b>
		Default Value: a100h Format: U16
646..647	63:48	<b>Forward R-ch Gamma Corrected Value 161</b>
		Default Value: a100h Format: U16
	47:32	<b>Forward Pixel Value 161</b>
		Default Value: a100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 161</b>
		Default Value: a100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 161</b>
		Default Value: a100h Format: U16
648..649	63:48	<b>Inverse R-ch Gamma Corrected Value 162</b>
		Default Value: a200h Format: U16
	47:32	<b>Inverse Pixel Value 162</b>
		Default Value: a200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 162</b>
		Default Value: a200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 162</b>
		Default Value: a200h Format: U16
650..651	63:48	<b>Forward R-ch Gamma Corrected Value 162</b>
		Default Value: a200h Format: U16
	47:32	<b>Forward Pixel Value 162</b>
		Default Value: a200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 162</b>
		Default Value: a200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 162</b>
		Default Value: a200h Format: U16
652..653	63:48	<b>Inverse R-ch Gamma Corrected Value 163</b>
		Default Value: a300h Format: U16
	47:32	<b>Inverse Pixel Value 163</b>
		Default Value: a300h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 163</b>
		Default Value: a300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 163</b>
		Default Value: a300h Format: U16
654..655	63:48	<b>Forward R-ch Gamma Corrected Value 163</b>
		Default Value: a300h Format: U16
	47:32	<b>Forward Pixel Value 163</b>
		Default Value: a300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 163</b>
		Default Value: a300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 163</b>
		Default Value: a300h Format: U16
656..657	63:48	<b>Inverse R-ch Gamma Corrected Value 164</b>
		Default Value: a400h Format: U16
	47:32	<b>Inverse Pixel Value 164</b>
		Default Value: a400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 164</b>
		Default Value: a400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 164</b>
		Default Value: a400h Format: U16
658..659	63:48	<b>Forward R-ch Gamma Corrected Value 164</b>
		Default Value: a400h Format: U16
	47:32	<b>Forward Pixel Value 164</b>
		Default Value: a400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 164</b>
		Default Value: a400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 164</b>
		Default Value: a400h Format: U16
660..661	63:48	<b>Inverse R-ch Gamma Corrected Value 165</b>
		Default Value: a500h Format: U16
	47:32	<b>Inverse Pixel Value 165</b>
		Default Value: a500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 165</b>
		Default Value: a500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 165</b>
		Default Value: a500h Format: U16
662..663	63:48	<b>Forward R-ch Gamma Corrected Value 165</b>
		Default Value: a500h Format: U16
	47:32	<b>Forward Pixel Value 165</b>
		Default Value: a500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 165</b>
		Default Value: a500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 165</b>
		Default Value: a500h Format: U16
664..665	63:48	<b>Inverse R-ch Gamma Corrected Value 166</b>
		Default Value: a600h Format: U16
	47:32	<b>Inverse Pixel Value 166</b>
		Default Value: a600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 166</b>
		Default Value: a600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 166</b>
		Default Value: a600h Format: U16
666..667	63:48	<b>Forward R-ch Gamma Corrected Value 166</b>
		Default Value: a600h Format: U16
	47:32	<b>Forward Pixel Value 166</b>
		Default Value: a600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 166</b>
		Default Value: a600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 166</b>
		Default Value: a600h Format: U16
668..669	63:48	<b>Inverse R-ch Gamma Corrected Value 167</b>
		Default Value: a700h Format: U16
	47:32	<b>Inverse Pixel Value 167</b>
		Default Value: a700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 167</b>
		Default Value: a700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 167</b>
		Default Value: a700h Format: U16
670..671	63:48	<b>Forward R-ch Gamma Corrected Value 167</b>
		Default Value: a700h Format: U16
	47:32	<b>Forward Pixel Value 167</b>
		Default Value: a700h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 167</b>
		Default Value: a700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 167</b>
		Default Value: a700h Format: U16
672..673	63:48	<b>Inverse R-ch Gamma Corrected Value 168</b>
		Default Value: a800h Format: U16
	47:32	<b>Inverse Pixel Value 168</b>
		Default Value: a800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 168</b>
		Default Value: a800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 168</b>
		Default Value: a800h Format: U16
674..675	63:48	<b>Forward R-ch Gamma Corrected Value 168</b>
		Default Value: a800h Format: U16
	47:32	<b>Forward Pixel Value 168</b>
		Default Value: a800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 168</b>
		Default Value: a800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 168</b>
		Default Value: a800h Format: U16
676..677	63:48	<b>Inverse R-ch Gamma Corrected Value 169</b>
		Default Value: a900h Format: U16
	47:32	<b>Inverse Pixel Value 169</b>
		Default Value: a900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 169</b>
		Default Value: a900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 169</b>
		Default Value: a900h Format: U16
678..679	63:48	<b>Forward R-ch Gamma Corrected Value 169</b>
		Default Value: a900h Format: U16
	47:32	<b>Forward Pixel Value 169</b>
		Default Value: a900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 169</b>
		Default Value: a900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 169</b>
		Default Value: a900h Format: U16
680..681	63:48	<b>Inverse R-ch Gamma Corrected Value 170</b>
		Default Value: aa00h Format: U16
	47:32	<b>Inverse Pixel Value 170</b>
		Default Value: aa00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 170</b>
		Default Value: aa00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 170</b>
		Default Value: aa00h Format: U16
682..683	63:48	<b>Forward R-ch Gamma Corrected Value 170</b>
		Default Value: aa00h Format: U16
	47:32	<b>Forward Pixel Value 170</b>
		Default Value: aa00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 170</b>
		Default Value: aa00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 170</b>
		Default Value: aa00h Format: U16
684..685	63:48	<b>Inverse R-ch Gamma Corrected Value 171</b>
		Default Value: ab00h Format: U16
	47:32	<b>Inverse Pixel Value 171</b>
		Default Value: ab00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 171</b>
		Default Value: ab00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 171</b>
		Default Value: ab00h Format: U16
686..687	63:48	<b>Forward R-ch Gamma Corrected Value 171</b>
		Default Value: ab00h Format: U16
	47:32	<b>Forward Pixel Value 171</b>
		Default Value: ab00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 171</b>
		Default Value: ab00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 171</b>
		Default Value: ab00h Format: U16
688..689	63:48	<b>Inverse R-ch Gamma Corrected Value 172</b>
		Default Value: ac00h Format: U16
	47:32	<b>Inverse Pixel Value 172</b>
		Default Value: ac00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 172</b>
		Default Value: ac00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 172</b>
		Default Value: ac00h Format: U16
690..691	63:48	<b>Forward R-ch Gamma Corrected Value 172</b>
		Default Value: ac00h Format: U16
	47:32	<b>Forward Pixel Value 172</b>
		Default Value: ac00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 172</b>
		Default Value: ac00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 172</b>
		Default Value: ac00h Format: U16
692..693	63:48	<b>Inverse R-ch Gamma Corrected Value 173</b>
		Default Value: ad00h Format: U16
	47:32	<b>Inverse Pixel Value 173</b>
		Default Value: ad00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 173</b>
		Default Value: ad00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 173</b>
		Default Value: ad00h Format: U16
694..695	63:48	<b>Forward R-ch Gamma Corrected Value 173</b>
		Default Value: ad00h Format: U16
	47:32	<b>Forward Pixel Value 173</b>
		Default Value: ad00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 173</b>
		Default Value: ad00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 173</b>
		Default Value: ad00h Format: U16
696..697	63:48	<b>Inverse R-ch Gamma Corrected Value 174</b>
		Default Value: ae00h Format: U16
	47:32	<b>Inverse Pixel Value 174</b>
		Default Value: ae00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 174</b>
		Default Value: ae00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 174</b>
		Default Value: ae00h Format: U16
698..699	63:48	<b>Forward R-ch Gamma Corrected Value 174</b>
		Default Value: ae00h Format: U16
	47:32	<b>Forward Pixel Value 174</b>
		Default Value: ae00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 174</b>
		Default Value: ae00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 174</b>
		Default Value: ae00h Format: U16
700..701	63:48	<b>Inverse R-ch Gamma Corrected Value 175</b>
		Default Value: af00h Format: U16
	47:32	<b>Inverse Pixel Value 175</b>
		Default Value: af00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 175</b>
		Default Value: af00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 175</b>
		Default Value: af00h Format: U16
702..703	63:48	<b>Forward R-ch Gamma Corrected Value 175</b>
		Default Value: af00h Format: U16
	47:32	<b>Forward Pixel Value 175</b>
		Default Value: af00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 175</b>
		Default Value: af00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 175</b>
		Default Value: af00h Format: U16
704..705	63:48	<b>Inverse R-ch Gamma Corrected Value 176</b>
		Default Value: b000h Format: U16
	47:32	<b>Inverse Pixel Value 176</b>
		Default Value: b000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 176</b>
		Default Value: b000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 176</b>
		Default Value: b000h Format: U16
706..707	63:48	<b>Forward R-ch Gamma Corrected Value 176</b>
		Default Value: b000h Format: U16
	47:32	<b>Forward Pixel Value 176</b>
		Default Value: b000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 176</b>
		Default Value: b000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 176</b>
		Default Value: b000h Format: U16
708..709	63:48	<b>Inverse R-ch Gamma Corrected Value 177</b>
		Default Value: b100h Format: U16
	47:32	<b>Inverse Pixel Value 177</b>
		Default Value: b100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 177</b>
		Default Value: b100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 177</b>
		Default Value: b100h Format: U16
710..711	63:48	<b>Forward R-ch Gamma Corrected Value 177</b>
		Default Value: b100h Format: U16
	47:32	<b>Forward Pixel Value 177</b>
		Default Value: b100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 177</b>
		Default Value: b100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 177</b>
		Default Value: b100h Format: U16
712..713	63:48	<b>Inverse R-ch Gamma Corrected Value 178</b>
		Default Value: b200h Format: U16
	47:32	<b>Inverse Pixel Value 178</b>
		Default Value: b200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 178</b>
		Default Value: b200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 178</b>
		Default Value: b200h Format: U16
714..715	63:48	<b>Forward R-ch Gamma Corrected Value 178</b>
		Default Value: b200h Format: U16
	47:32	<b>Forward Pixel Value 178</b>
		Default Value: b200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 178</b>
		Default Value: b200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 178</b>
		Default Value: b200h Format: U16
716..717	63:48	<b>Inverse R-ch Gamma Corrected Value 179</b>
		Default Value: b300h Format: U16
	47:32	<b>Inverse Pixel Value 179</b>
		Default Value: b300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 179</b>
		Default Value: b300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 179</b>
		Default Value: b300h Format: U16
718..719	63:48	<b>Forward R-ch Gamma Corrected Value 179</b>
		Default Value: b300h Format: U16
	47:32	<b>Forward Pixel Value 179</b>
		Default Value: b300h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 179</b>
		Default Value: b300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 179</b>
		Default Value: b300h Format: U16
720..721	63:48	<b>Inverse R-ch Gamma Corrected Value 180</b>
		Default Value: b400h Format: U16
	47:32	<b>Inverse Pixel Value 180</b>
		Default Value: b400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 180</b>
		Default Value: b400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 180</b>
		Default Value: b400h Format: U16
722..723	63:48	<b>Forward R-ch Gamma Corrected Value 180</b>
		Default Value: b400h Format: U16
	47:32	<b>Forward Pixel Value 180</b>
		Default Value: b400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 180</b>
		Default Value: b400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 180</b>
		Default Value: b400h Format: U16
724..725	63:48	<b>Inverse R-ch Gamma Corrected Value 181</b>
		Default Value: b500h Format: U16
	47:32	<b>Inverse Pixel Value 181</b>
		Default Value: b500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 181</b>
		Default Value: b500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 181</b>
		Default Value: b500h Format: U16
726..727	63:48	<b>Forward R-ch Gamma Corrected Value 181</b>
		Default Value: b500h Format: U16
	47:32	<b>Forward Pixel Value 181</b>
		Default Value: b500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 181</b>
		Default Value: b500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 181</b>
		Default Value: b500h Format: U16
728..729	63:48	<b>Inverse R-ch Gamma Corrected Value 182</b>
		Default Value: b600h Format: U16
	47:32	<b>Inverse Pixel Value 182</b>
		Default Value: b600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 182</b>
		Default Value: b600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 182</b>
		Default Value: b600h Format: U16
730..731	63:48	<b>Forward R-ch Gamma Corrected Value 182</b>
		Default Value: b600h Format: U16
	47:32	<b>Forward Pixel Value 182</b>
		Default Value: b600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 182</b>
		Default Value: b600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 182</b>
		Default Value: b600h Format: U16
732..733	63:48	<b>Inverse R-ch Gamma Corrected Value 183</b>
		Default Value: b700h Format: U16
	47:32	<b>Inverse Pixel Value 183</b>
		Default Value: b700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 183</b>
		Default Value: b700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 183</b>
		Default Value: b700h Format: U16
734..735	63:48	<b>Forward R-ch Gamma Corrected Value 183</b>
		Default Value: b700h Format: U16
	47:32	<b>Forward Pixel Value 183</b>
		Default Value: b700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 183</b>
		Default Value: b700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 183</b>
		Default Value: b700h Format: U16
736..737	63:48	<b>Inverse R-ch Gamma Corrected Value 184</b>
		Default Value: b800h Format: U16
	47:32	<b>Inverse Pixel Value 184</b>
		Default Value: b800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 184</b>
		Default Value: b800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 184</b>
		Default Value: b800h Format: U16
738..739	63:48	<b>Forward R-ch Gamma Corrected Value 184</b>
		Default Value: b800h Format: U16
	47:32	<b>Forward Pixel Value 184</b>
		Default Value: b800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 184</b>
		Default Value: b800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 184</b>
		Default Value: b800h Format: U16
740..741	63:48	<b>Inverse R-ch Gamma Corrected Value 185</b>
		Default Value: b900h Format: U16
	47:32	<b>Inverse Pixel Value 185</b>
		Default Value: b900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 185</b>
		Default Value: b900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 185</b>
		Default Value: b900h Format: U16
742..743	63:48	<b>Forward R-ch Gamma Corrected Value 185</b>
		Default Value: b900h Format: U16
	47:32	<b>Forward Pixel Value 185</b>
		Default Value: b900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 185</b>
		Default Value: b900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 185</b>
		Default Value: b900h Format: U16
744..745	63:48	<b>Inverse R-ch Gamma Corrected Value 186</b>
		Default Value: ba00h Format: U16
	47:32	<b>Inverse Pixel Value 186</b>
		Default Value: ba00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 186</b>
		Default Value: ba00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 186</b>
		Default Value: ba00h Format: U16
746..747	63:48	<b>Forward R-ch Gamma Corrected Value 186</b>
		Default Value: ba00h Format: U16
	47:32	<b>Forward Pixel Value 186</b>
		Default Value: ba00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 186</b>
		Default Value: ba00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 186</b>
		Default Value: ba00h Format: U16
748..749	63:48	<b>Inverse R-ch Gamma Corrected Value 187</b>
		Default Value: bb00h Format: U16
	47:32	<b>Inverse Pixel Value 187</b>
		Default Value: bb00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 187</b>
		Default Value: bb00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 187</b>
		Default Value: bb00h Format: U16
750..751	63:48	<b>Forward R-ch Gamma Corrected Value 187</b>
		Default Value: bb00h Format: U16
	47:32	<b>Forward Pixel Value 187</b>
		Default Value: bb00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 187</b>
		Default Value: bb00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 187</b>
		Default Value: bb00h Format: U16
752..753	63:48	<b>Inverse R-ch Gamma Corrected Value 188</b>
		Default Value: bc00h Format: U16
	47:32	<b>Inverse Pixel Value 188</b>
		Default Value: bc00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 188</b>
		Default Value: bc00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 188</b>
		Default Value: bc00h Format: U16
754..755	63:48	<b>Forward R-ch Gamma Corrected Value 188</b>
		Default Value: bc00h Format: U16
	47:32	<b>Forward Pixel Value 188</b>
		Default Value: bc00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 188</b>
		Default Value: bc00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 188</b>
		Default Value: bc00h Format: U16
756..757	63:48	<b>Inverse R-ch Gamma Corrected Value 189</b>
		Default Value: bd00h Format: U16
	47:32	<b>Inverse Pixel Value 189</b>
		Default Value: bd00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 189</b>
		Default Value: bd00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 189</b>
		Default Value: bd00h Format: U16
758..759	63:48	<b>Forward R-ch Gamma Corrected Value 189</b>
		Default Value: bd00h Format: U16
	47:32	<b>Forward Pixel Value 189</b>
		Default Value: bd00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 189</b>
		Default Value: bd00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 189</b>
		Default Value: bd00h Format: U16
760..761	63:48	<b>Inverse R-ch Gamma Corrected Value 190</b>
		Default Value: be00h Format: U16
	47:32	<b>Inverse Pixel Value 190</b>
		Default Value: be00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 190</b>
		Default Value: be00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 190</b>
		Default Value: be00h Format: U16
762..763	63:48	<b>Forward R-ch Gamma Corrected Value 190</b>
		Default Value: be00h Format: U16
	47:32	<b>Forward Pixel Value 190</b>
		Default Value: be00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 190</b>
		Default Value: be00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 190</b>
		Default Value: be00h Format: U16
764..765	63:48	<b>Inverse R-ch Gamma Corrected Value 191</b>
		Default Value: bf00h Format: U16
	47:32	<b>Inverse Pixel Value 191</b>
		Default Value: bf00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 191</b>
		Default Value: bf00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 191</b>
		Default Value: bf00h Format: U16
766..767	63:48	<b>Forward R-ch Gamma Corrected Value 191</b>
		Default Value: bf00h Format: U16
	47:32	<b>Forward Pixel Value 191</b>
		Default Value: bf00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 191</b>
		Default Value: bf00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 191</b>
		Default Value: bf00h Format: U16
768..769	63:48	<b>Inverse R-ch Gamma Corrected Value 192</b>
		Default Value: c000h Format: U16
	47:32	<b>Inverse Pixel Value 192</b>
		Default Value: c000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 192</b>
		Default Value: c000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 192</b>
		Default Value: c000h Format: U16
770..771	63:48	<b>Forward R-ch Gamma Corrected Value 192</b>
		Default Value: c000h Format: U16
	47:32	<b>Forward Pixel Value 192</b>
		Default Value: c000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 192</b>
		Default Value: c000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 192</b>
		Default Value: c000h Format: U16
772..773	63:48	<b>Inverse R-ch Gamma Corrected Value 193</b>
		Default Value: c100h Format: U16
	47:32	<b>Inverse Pixel Value 193</b>
		Default Value: c100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 193</b>
		Default Value: c100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 193</b>
		Default Value: c100h Format: U16
774..775	63:48	<b>Forward R-ch Gamma Corrected Value 193</b>
		Default Value: c100h Format: U16
	47:32	<b>Forward Pixel Value 193</b>
		Default Value: c100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 193</b>
		Default Value: c100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 193</b>
		Default Value: c100h Format: U16
776..777	63:48	<b>Inverse R-ch Gamma Corrected Value 194</b>
		Default Value: c200h Format: U16
	47:32	<b>Inverse Pixel Value 194</b>
		Default Value: c200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 194</b>
		Default Value: c200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 194</b>
		Default Value: c200h Format: U16
778..779	63:48	<b>Forward R-ch Gamma Corrected Value 194</b>
		Default Value: c200h Format: U16
	47:32	<b>Forward Pixel Value 194</b>
		Default Value: c200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 194</b>
		Default Value: c200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 194</b>
		Default Value: c200h Format: U16
780..781	63:48	<b>Inverse R-ch Gamma Corrected Value 195</b>
		Default Value: c300h Format: U16
	47:32	<b>Inverse Pixel Value 195</b>
		Default Value: c300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 195</b>
		Default Value: c300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 195</b>
		Default Value: c300h Format: U16
782..783	63:48	<b>Forward R-ch Gamma Corrected Value 195</b>
		Default Value: c300h Format: U16
	47:32	<b>Forward Pixel Value 195</b>
		Default Value: c300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 195</b>
		Default Value: c300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 195</b>
		Default Value: c300h Format: U16
784..785	63:48	<b>Inverse R-ch Gamma Corrected Value 196</b>
		Default Value: c400h Format: U16
	47:32	<b>Inverse Pixel Value 196</b>
		Default Value: c400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 196</b>
		Default Value: c400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 196</b>
		Default Value: c400h Format: U16
786..787	63:48	<b>Forward R-ch Gamma Corrected Value 196</b>
		Default Value: c400h Format: U16
	47:32	<b>Forward Pixel Value 196</b>
		Default Value: c400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 196</b>
		Default Value: c400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 196</b>
		Default Value: c400h Format: U16
788..789	63:48	<b>Inverse R-ch Gamma Corrected Value 197</b>
		Default Value: c500h Format: U16
	47:32	<b>Inverse Pixel Value 197</b>
		Default Value: c500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 197</b>
		Default Value: c500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 197</b>
		Default Value: c500h Format: U16
790..791	63:48	<b>Forward R-ch Gamma Corrected Value 197</b>
		Default Value: c500h Format: U16
	47:32	<b>Forward Pixel Value 197</b>
		Default Value: c500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 197</b>
		Default Value: c500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 197</b>
		Default Value: c500h Format: U16
792..793	63:48	<b>Inverse R-ch Gamma Corrected Value 198</b>
		Default Value: c600h Format: U16
	47:32	<b>Inverse Pixel Value 198</b>
		Default Value: c600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 198</b>
		Default Value: c600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 198</b>
		Default Value: c600h Format: U16
794..795	63:48	<b>Forward R-ch Gamma Corrected Value 198</b>
		Default Value: c600h Format: U16
	47:32	<b>Forward Pixel Value 198</b>
		Default Value: c600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 198</b>
		Default Value: c600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 198</b>
		Default Value: c600h Format: U16
796..797	63:48	<b>Inverse R-ch Gamma Corrected Value 199</b>
		Default Value: c700h Format: U16
	47:32	<b>Inverse Pixel Value 199</b>
		Default Value: c700h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 199</b>
		Default Value: c700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 199</b>
		Default Value: c700h Format: U16
798..799	63:48	<b>Forward R-ch Gamma Corrected Value 199</b>
		Default Value: c700h Format: U16
	47:32	<b>Forward Pixel Value 199</b>
		Default Value: c700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 199</b>
		Default Value: c700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 199</b>
		Default Value: c700h Format: U16
800..801	63:48	<b>Inverse R-ch Gamma Corrected Value 200</b>
		Default Value: c800h Format: U16
	47:32	<b>Inverse Pixel Value 200</b>
		Default Value: c800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 200</b>
		Default Value: c800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 200</b>
		Default Value: c800h Format: U16
802..803	63:48	<b>Forward R-ch Gamma Corrected Value 200</b>
		Default Value: c800h Format: U16
	47:32	<b>Forward Pixel Value 200</b>
		Default Value: c800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 200</b>
		Default Value: c800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 200</b>
		Default Value: c800h Format: U16
804..805	63:48	<b>Inverse R-ch Gamma Corrected Value 201</b>
		Default Value: c900h Format: U16
	47:32	<b>Inverse Pixel Value 201</b>
		Default Value: c900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 201</b>
		Default Value: c900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 201</b>
		Default Value: c900h Format: U16
806..807	63:48	<b>Forward R-ch Gamma Corrected Value 201</b>
		Default Value: c900h Format: U16
	47:32	<b>Forward Pixel Value 201</b>
		Default Value: c900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 201</b>
		Default Value: c900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 201</b>
		Default Value: c900h Format: U16
808..809	63:48	<b>Inverse R-ch Gamma Corrected Value 202</b>
		Default Value: ca00h Format: U16
	47:32	<b>Inverse Pixel Value 202</b>
		Default Value: ca00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 202</b>
		Default Value: ca00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 202</b>
		Default Value: ca00h Format: U16
810..811	63:48	<b>Forward R-ch Gamma Corrected Value 202</b>
		Default Value: ca00h Format: U16
	47:32	<b>Forward Pixel Value 202</b>
		Default Value: ca00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 202</b>
		Default Value: ca00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 202</b>
		Default Value: ca00h Format: U16
812..813	63:48	<b>Inverse R-ch Gamma Corrected Value 203</b>
		Default Value: cb00h Format: U16
	47:32	<b>Inverse Pixel Value 203</b>
		Default Value: cb00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 203</b>
		Default Value: cb00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 203</b>
		Default Value: cb00h Format: U16
814..815	63:48	<b>Forward R-ch Gamma Corrected Value 203</b>
		Default Value: cb00h Format: U16
	47:32	<b>Forward Pixel Value 203</b>
		Default Value: cb00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 203</b>
		Default Value: cb00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 203</b>
		Default Value: cb00h Format: U16
816..817	63:48	<b>Inverse R-ch Gamma Corrected Value 204</b>
		Default Value: cc00h Format: U16
	47:32	<b>Inverse Pixel Value 204</b>
		Default Value: cc00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 204</b>
		Default Value: cc00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 204</b>
		Default Value: cc00h Format: U16
818..819	63:48	<b>Forward R-ch Gamma Corrected Value 204</b>
		Default Value: cc00h Format: U16
	47:32	<b>Forward Pixel Value 204</b>
		Default Value: cc00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 204</b>
		Default Value: cc00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 204</b>
		Default Value: cc00h Format: U16
820..821	63:48	<b>Inverse R-ch Gamma Corrected Value 205</b>
		Default Value: cd00h Format: U16
	47:32	<b>Inverse Pixel Value 205</b>
		Default Value: cd00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 205</b>
		Default Value: cd00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 205</b>
		Default Value: cd00h Format: U16
822..823	63:48	<b>Forward R-ch Gamma Corrected Value 205</b>
		Default Value: cd00h Format: U16
	47:32	<b>Forward Pixel Value 205</b>
		Default Value: cd00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 205</b>
		Default Value: cd00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 205</b>
		Default Value: cd00h Format: U16
824..825	63:48	<b>Inverse R-ch Gamma Corrected Value 206</b>
		Default Value: ce00h Format: U16
	47:32	<b>Inverse Pixel Value 206</b>
		Default Value: ce00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 206</b>
		Default Value: ce00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 206</b>
		Default Value: ce00h Format: U16
826..827	63:48	<b>Forward R-ch Gamma Corrected Value 206</b>
		Default Value: ce00h Format: U16
	47:32	<b>Forward Pixel Value 206</b>
		Default Value: ce00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 206</b>
		Default Value: ce00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 206</b>
		Default Value: ce00h Format: U16
828..829	63:48	<b>Inverse R-ch Gamma Corrected Value 207</b>
		Default Value: cf00h Format: U16
	47:32	<b>Inverse Pixel Value 207</b>
		Default Value: cf00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 207</b>
		Default Value: cf00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 207</b>
		Default Value: cf00h Format: U16
830..831	63:48	<b>Forward R-ch Gamma Corrected Value 207</b>
		Default Value: cf00h Format: U16
	47:32	<b>Forward Pixel Value 207</b>
		Default Value: cf00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 207</b>
		Default Value: cf00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 207</b>
		Default Value: cf00h Format: U16
832..833	63:48	<b>Inverse R-ch Gamma Corrected Value 208</b>
		Default Value: d000h Format: U16
	47:32	<b>Inverse Pixel Value 208</b>
		Default Value: d000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 208</b>
		Default Value: d000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 208</b>
		Default Value: d000h Format: U16
834..835	63:48	<b>Forward R-ch Gamma Corrected Value 208</b>
		Default Value: d000h Format: U16
	47:32	<b>Forward Pixel Value 208</b>
		Default Value: d000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 208</b>
		Default Value: d000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 208</b>
		Default Value: d000h Format: U16
836..837	63:48	<b>Inverse R-ch Gamma Corrected Value 209</b>
		Default Value: d100h Format: U16
	47:32	<b>Inverse Pixel Value 209</b>
		Default Value: d100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 209</b>
		Default Value: d100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 209</b>
		Default Value: d100h Format: U16
838..839	63:48	<b>Forward R-ch Gamma Corrected Value 209</b>
		Default Value: d100h Format: U16
	47:32	<b>Forward Pixel Value 209</b>
		Default Value: d100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 209</b>
		Default Value: d100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 209</b>
		Default Value: d100h Format: U16
840..841	63:48	<b>Inverse R-ch Gamma Corrected Value 210</b>
		Default Value: d200h Format: U16
	47:32	<b>Inverse Pixel Value 210</b>
		Default Value: d200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 210</b>
		Default Value: d200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 210</b>
		Default Value: d200h Format: U16
842..843	63:48	<b>Forward R-ch Gamma Corrected Value 210</b>
		Default Value: d200h Format: U16
	47:32	<b>Forward Pixel Value 210</b>
		Default Value: d200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 210</b>
		Default Value: d200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 210</b>
		Default Value: d200h Format: U16
844..845	63:48	<b>Inverse R-ch Gamma Corrected Value 211</b>
		Default Value: d300h Format: U16
	47:32	<b>Inverse Pixel Value 211</b>
		Default Value: d300h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 211</b>
		Default Value: d300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 211</b>
		Default Value: d300h Format: U16
846..847	63:48	<b>Forward R-ch Gamma Corrected Value 211</b>
		Default Value: d300h Format: U16
	47:32	<b>Forward Pixel Value 211</b>
		Default Value: d300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 211</b>
		Default Value: d300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 211</b>
		Default Value: d300h Format: U16
848..849	63:48	<b>Inverse R-ch Gamma Corrected Value 212</b>
		Default Value: d400h Format: U16
	47:32	<b>Inverse Pixel Value 212</b>
		Default Value: d400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 212</b>
		Default Value: d400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 212</b>
		Default Value: d400h Format: U16
850..851	63:48	<b>Forward R-ch Gamma Corrected Value 212</b>
		Default Value: d400h Format: U16
	47:32	<b>Forward Pixel Value 212</b>
		Default Value: d400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 212</b>
		Default Value: d400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 212</b>
		Default Value: d400h Format: U16
852..853	63:48	<b>Inverse R-ch Gamma Corrected Value 213</b>
		Default Value: d500h Format: U16
	47:32	<b>Inverse Pixel Value 213</b>
		Default Value: d500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 213</b>
		Default Value: d500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 213</b>
		Default Value: d500h Format: U16
854..855	63:48	<b>Forward R-ch Gamma Corrected Value 213</b>
		Default Value: d500h Format: U16
	47:32	<b>Forward Pixel Value 213</b>
		Default Value: d500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 213</b>
		Default Value: d500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 213</b>
		Default Value: d500h Format: U16
856..857	63:48	<b>Inverse R-ch Gamma Corrected Value 214</b>
		Default Value: d600h Format: U16
	47:32	<b>Inverse Pixel Value 214</b>
		Default Value: d600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 214</b>
		Default Value: d600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 214</b>
		Default Value: d600h Format: U16
858..859	63:48	<b>Forward R-ch Gamma Corrected Value 214</b>
		Default Value: d600h Format: U16
	47:32	<b>Forward Pixel Value 214</b>
		Default Value: d600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 214</b>
		Default Value: d600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 214</b>
		Default Value: d600h Format: U16
860..861	63:48	<b>Inverse R-ch Gamma Corrected Value 215</b>
		Default Value: d700h Format: U16
	47:32	<b>Inverse Pixel Value 215</b>
		Default Value: d700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 215</b>
		Default Value: d700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 215</b>
		Default Value: d700h Format: U16
862..863	63:48	<b>Forward R-ch Gamma Corrected Value 215</b>
		Default Value: d700h Format: U16
	47:32	<b>Forward Pixel Value 215</b>
		Default Value: d700h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 215</b>
		Default Value: d700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 215</b>
		Default Value: d700h Format: U16
864..865	63:48	<b>Inverse R-ch Gamma Corrected Value 216</b>
		Default Value: d800h Format: U16
	47:32	<b>Inverse Pixel Value 216</b>
		Default Value: d800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 216</b>
		Default Value: d800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 216</b>
		Default Value: d800h Format: U16
866..867	63:48	<b>Forward R-ch Gamma Corrected Value 216</b>
		Default Value: d800h Format: U16
	47:32	<b>Forward Pixel Value 216</b>
		Default Value: d800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 216</b>
		Default Value: d800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 216</b>
		Default Value: d800h Format: U16
868..869	63:48	<b>Inverse R-ch Gamma Corrected Value 217</b>
		Default Value: d900h Format: U16
	47:32	<b>Inverse Pixel Value 217</b>
		Default Value: d900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 217</b>
		Default Value: d900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 217</b>
		Default Value: d900h Format: U16
870..871	63:48	<b>Forward R-ch Gamma Corrected Value 217</b>
		Default Value: d900h Format: U16
	47:32	<b>Forward Pixel Value 217</b>
		Default Value: d900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 217</b>
		Default Value: d900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 217</b>
		Default Value: d900h Format: U16
872..873	63:48	<b>Inverse R-ch Gamma Corrected Value 218</b>
		Default Value: da00h Format: U16
	47:32	<b>Inverse Pixel Value 218</b>
		Default Value: da00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 218</b>
		Default Value: da00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 218</b>
		Default Value: da00h Format: U16
874..875	63:48	<b>Forward R-ch Gamma Corrected Value 218</b>
		Default Value: da00h Format: U16
	47:32	<b>Forward Pixel Value 218</b>
		Default Value: da00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 218</b>
		Default Value: da00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 218</b>
		Default Value: da00h Format: U16
876..877	63:48	<b>Inverse R-ch Gamma Corrected Value 219</b>
		Default Value: db00h Format: U16
	47:32	<b>Inverse Pixel Value 219</b>
		Default Value: db00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 219</b>
		Default Value: db00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 219</b>
		Default Value: db00h Format: U16
878..879	63:48	<b>Forward R-ch Gamma Corrected Value 219</b>
		Default Value: db00h Format: U16
	47:32	<b>Forward Pixel Value 219</b>
		Default Value: db00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 219</b>
		Default Value: db00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 219</b>
		Default Value: db00h Format: U16
880..881	63:48	<b>Inverse R-ch Gamma Corrected Value 220</b>
		Default Value: dc00h Format: U16
	47:32	<b>Inverse Pixel Value 220</b>
		Default Value: dc00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 220</b>
		Default Value: dc00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 220</b>
		Default Value: dc00h Format: U16
882..883	63:48	<b>Forward R-ch Gamma Corrected Value 220</b>
		Default Value: dc00h Format: U16
	47:32	<b>Forward Pixel Value 220</b>
		Default Value: dc00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 220</b>
		Default Value: dc00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 220</b>
		Default Value: dc00h Format: U16
884..885	63:48	<b>Inverse R-ch Gamma Corrected Value 221</b>
		Default Value: dd00h Format: U16
	47:32	<b>Inverse Pixel Value 221</b>
		Default Value: dd00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 221</b>
		Default Value: dd00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 221</b>
		Default Value: dd00h Format: U16
886..887	63:48	<b>Forward R-ch Gamma Corrected Value 221</b>
		Default Value: dd00h Format: U16
	47:32	<b>Forward Pixel Value 221</b>
		Default Value: dd00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 221</b>
		Default Value: dd00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 221</b>
		Default Value: dd00h Format: U16
888..889	63:48	<b>Inverse R-ch Gamma Corrected Value 222</b>
		Default Value: de00h Format: U16
	47:32	<b>Inverse Pixel Value 222</b>
		Default Value: de00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 222</b>
		Default Value: de00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 222</b>
		Default Value: de00h Format: U16
890..891	63:48	<b>Forward R-ch Gamma Corrected Value 222</b>
		Default Value: de00h Format: U16
	47:32	<b>Forward Pixel Value 222</b>
		Default Value: de00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 222</b>
		Default Value: de00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 222</b>
		Default Value: de00h Format: U16
892..893	63:48	<b>Inverse R-ch Gamma Corrected Value 223</b>
		Default Value: df00h Format: U16
	47:32	<b>Inverse Pixel Value 223</b>
		Default Value: df00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 223</b>
		Default Value: df00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 223</b>
		Default Value: df00h Format: U16
894..895	63:48	<b>Forward R-ch Gamma Corrected Value 223</b>
		Default Value: df00h Format: U16
	47:32	<b>Forward Pixel Value 223</b>
		Default Value: df00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 223</b>
		Default Value: df00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 223</b>
		Default Value: df00h Format: U16
896..897	63:48	<b>Inverse R-ch Gamma Corrected Value 224</b>
		Default Value: e000h Format: U16
	47:32	<b>Inverse Pixel Value 224</b>
		Default Value: e000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 224</b>
		Default Value: e000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 224</b>
		Default Value: e000h Format: U16
898..899	63:48	<b>Forward R-ch Gamma Corrected Value 224</b>
		Default Value: e000h Format: U16
	47:32	<b>Forward Pixel Value 224</b>
		Default Value: e000h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 224</b>
		Default Value: e000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 224</b>
		Default Value: e000h Format: U16
900..901	63:48	<b>Inverse R-ch Gamma Corrected Value 225</b>
		Default Value: e100h Format: U16
	47:32	<b>Inverse Pixel Value 225</b>
		Default Value: e100h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 225</b>
		Default Value: e100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 225</b>
		Default Value: e100h Format: U16
902..903	63:48	<b>Forward R-ch Gamma Corrected Value 225</b>
		Default Value: e100h Format: U16
	47:32	<b>Forward Pixel Value 225</b>
		Default Value: e100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 225</b>
		Default Value: e100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 225</b>
		Default Value: e100h Format: U16
904..905	63:48	<b>Inverse R-ch Gamma Corrected Value 226</b>
		Default Value: e200h Format: U16
	47:32	<b>Inverse Pixel Value 226</b>
		Default Value: e200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 226</b>
		Default Value: e200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 226</b>
		Default Value: e200h Format: U16
906..907	63:48	<b>Forward R-ch Gamma Corrected Value 226</b>
		Default Value: e200h Format: U16
	47:32	<b>Forward Pixel Value 226</b>
		Default Value: e200h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 226</b>
		Default Value: e200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 226</b>
		Default Value: e200h Format: U16
908..909	63:48	<b>Inverse R-ch Gamma Corrected Value 227</b>
		Default Value: e300h Format: U16
	47:32	<b>Inverse Pixel Value 227</b>
		Default Value: e300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 227</b>
		Default Value: e300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 227</b>
		Default Value: e300h Format: U16
910..911	63:48	<b>Forward R-ch Gamma Corrected Value 227</b>
		Default Value: e300h Format: U16
	47:32	<b>Forward Pixel Value 227</b>
		Default Value: e300h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 227</b>
		Default Value: e300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 227</b>
		Default Value: e300h Format: U16
912..913	63:48	<b>Inverse R-ch Gamma Corrected Value 228</b>
		Default Value: e400h Format: U16
	47:32	<b>Inverse Pixel Value 228</b>
		Default Value: e400h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 228</b>
		Default Value: e400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 228</b>
		Default Value: e400h Format: U16
914..915	63:48	<b>Forward R-ch Gamma Corrected Value 228</b>
		Default Value: e400h Format: U16
	47:32	<b>Forward Pixel Value 228</b>
		Default Value: e400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 228</b>
		Default Value: e400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 228</b>
		Default Value: e400h Format: U16
916..917	63:48	<b>Inverse R-ch Gamma Corrected Value 229</b>
		Default Value: e500h Format: U16
	47:32	<b>Inverse Pixel Value 229</b>
		Default Value: e500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 229</b>
		Default Value: e500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 229</b>
		Default Value: e500h Format: U16
918..919	63:48	<b>Forward R-ch Gamma Corrected Value 229</b>
		Default Value: e500h Format: U16
	47:32	<b>Forward Pixel Value 229</b>
		Default Value: e500h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 229</b>
		Default Value: e500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 229</b>
		Default Value: e500h Format: U16
920..921	63:48	<b>Inverse R-ch Gamma Corrected Value 230</b>
		Default Value: e600h Format: U16
	47:32	<b>Inverse Pixel Value 230</b>
		Default Value: e600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 230</b>
		Default Value: e600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 230</b>
		Default Value: e600h Format: U16
922..923	63:48	<b>Forward R-ch Gamma Corrected Value 230</b>
		Default Value: e600h Format: U16
	47:32	<b>Forward Pixel Value 230</b>
		Default Value: e600h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 230</b>
		Default Value: e600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 230</b>
		Default Value: e600h Format: U16
924..925	63:48	<b>Inverse R-ch Gamma Corrected Value 231</b>
		Default Value: e700h Format: U16
	47:32	<b>Inverse Pixel Value 231</b>
		Default Value: e700h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 231</b>
		Default Value: e700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 231</b>
		Default Value: e700h Format: U16
926..927	63:48	<b>Forward R-ch Gamma Corrected Value 231</b>
		Default Value: e700h Format: U16
	47:32	<b>Forward Pixel Value 231</b>
		Default Value: e700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 231</b>
		Default Value: e700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 231</b>
		Default Value: e700h Format: U16
928..929	63:48	<b>Inverse R-ch Gamma Corrected Value 232</b>
		Default Value: e800h Format: U16
	47:32	<b>Inverse Pixel Value 232</b>
		Default Value: e800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 232</b>
		Default Value: e800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 232</b>
		Default Value: e800h Format: U16
930..931	63:48	<b>Forward R-ch Gamma Corrected Value 232</b>
		Default Value: e800h Format: U16
	47:32	<b>Forward Pixel Value 232</b>
		Default Value: e800h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 232</b>
		Default Value: e800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 232</b>
		Default Value: e800h Format: U16
932..933	63:48	<b>Inverse R-ch Gamma Corrected Value 233</b>
		Default Value: e900h Format: U16
	47:32	<b>Inverse Pixel Value 233</b>
		Default Value: e900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 233</b>
		Default Value: e900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 233</b>
		Default Value: e900h Format: U16
934..935	63:48	<b>Forward R-ch Gamma Corrected Value 233</b>
		Default Value: e900h Format: U16
	47:32	<b>Forward Pixel Value 233</b>
		Default Value: e900h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 233</b>
		Default Value: e900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 233</b>
		Default Value: e900h Format: U16
936..937	63:48	<b>Inverse R-ch Gamma Corrected Value 234</b>
		Default Value: ea00h Format: U16
	47:32	<b>Inverse Pixel Value 234</b>
		Default Value: ea00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 234</b>
		Default Value: ea00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 234</b>
		Default Value: ea00h Format: U16
938..939	63:48	<b>Forward R-ch Gamma Corrected Value 234</b>
		Default Value: ea00h Format: U16
	47:32	<b>Forward Pixel Value 234</b>
		Default Value: ea00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 234</b>
		Default Value: ea00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 234</b>
		Default Value: ea00h Format: U16
940..941	63:48	<b>Inverse R-ch Gamma Corrected Value 235</b>
		Default Value: eb00h Format: U16
	47:32	<b>Inverse Pixel Value 235</b>
		Default Value: eb00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 235</b>
		Default Value: eb00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 235</b>
		Default Value: eb00h Format: U16
942..943	63:48	<b>Forward R-ch Gamma Corrected Value 235</b>
		Default Value: eb00h Format: U16
	47:32	<b>Forward Pixel Value 235</b>
		Default Value: eb00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 235</b>
		Default Value: eb00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 235</b>
		Default Value: eb00h Format: U16
944..945	63:48	<b>Inverse R-ch Gamma Corrected Value 236</b>
		Default Value: ec00h Format: U16
	47:32	<b>Inverse Pixel Value 236</b>
		Default Value: ec00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 236</b>
		Default Value: ec00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 236</b>
		Default Value: ec00h Format: U16
946..947	63:48	<b>Forward R-ch Gamma Corrected Value 236</b>
		Default Value: ec00h Format: U16
	47:32	<b>Forward Pixel Value 236</b>
		Default Value: ec00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 236</b>
		Default Value: ec00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 236</b>
		Default Value: ec00h Format: U16
948..949	63:48	<b>Inverse R-ch Gamma Corrected Value 237</b>
		Default Value: ed00h Format: U16
	47:32	<b>Inverse Pixel Value 237</b>
		Default Value: ed00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 237</b>
		Default Value: ed00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 237</b>
		Default Value: ed00h Format: U16
950..951	63:48	<b>Forward R-ch Gamma Corrected Value 237</b>
		Default Value: ed00h Format: U16
	47:32	<b>Forward Pixel Value 237</b>
		Default Value: ed00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 237</b>
		Default Value: ed00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 237</b>
		Default Value: ed00h Format: U16
952..953	63:48	<b>Inverse R-ch Gamma Corrected Value 238</b>
		Default Value: ee00h Format: U16
	47:32	<b>Inverse Pixel Value 238</b>
		Default Value: ee00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 238</b>
		Default Value: ee00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 238</b>
		Default Value: ee00h Format: U16
954..955	63:48	<b>Forward R-ch Gamma Corrected Value 238</b>
		Default Value: ee00h Format: U16
	47:32	<b>Forward Pixel Value 238</b>
		Default Value: ee00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 238</b>
		Default Value: ee00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 238</b>
		Default Value: ee00h Format: U16
956..957	63:48	<b>Inverse R-ch Gamma Corrected Value 239</b>
		Default Value: ef00h Format: U16
	47:32	<b>Inverse Pixel Value 239</b>
		Default Value: ef00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 239</b>
		Default Value: ef00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 239</b>
		Default Value: ef00h Format: U16
958..959	63:48	<b>Forward R-ch Gamma Corrected Value 239</b>
		Default Value: ef00h Format: U16
	47:32	<b>Forward Pixel Value 239</b>
		Default Value: ef00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 239</b>
		Default Value: ef00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 239</b>
		Default Value: ef00h Format: U16
960..961	63:48	<b>Inverse R-ch Gamma Corrected Value 240</b>
		Default Value: f000h Format: U16
	47:32	<b>Inverse Pixel Value 240</b>
		Default Value: f000h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 240</b>
		Default Value: f000h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 240</b>
		Default Value: f000h Format: U16
962..963	63:48	<b>Forward R-ch Gamma Corrected Value 240</b>
		Default Value: f000h Format: U16
	47:32	<b>Forward Pixel Value 240</b>
		Default Value: f000h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 240</b>
		Default Value: f000h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 240</b>
		Default Value: f000h Format: U16
964..965	63:48	<b>Inverse R-ch Gamma Corrected Value 241</b>
		Default Value: f100h Format: U16
	47:32	<b>Inverse Pixel Value 241</b>
		Default Value: f100h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 241</b>
		Default Value: f100h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 241</b>
		Default Value: f100h Format: U16
966..967	63:48	<b>Forward R-ch Gamma Corrected Value 241</b>
		Default Value: f100h Format: U16
	47:32	<b>Forward Pixel Value 241</b>
		Default Value: f100h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 241</b>
		Default Value: f100h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 241</b>
		Default Value: f100h Format: U16
968..969	63:48	<b>Inverse R-ch Gamma Corrected Value 242</b>
		Default Value: f200h Format: U16
	47:32	<b>Inverse Pixel Value 242</b>
		Default Value: f200h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 242</b>
		Default Value: f200h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 242</b>
		Default Value: f200h Format: U16
970..971	63:48	<b>Forward R-ch Gamma Corrected Value 242</b>
		Default Value: f200h Format: U16
	47:32	<b>Forward Pixel Value 242</b>
		Default Value: f200h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 242</b>
		Default Value: f200h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 242</b>
		Default Value: f200h Format: U16
972..973	63:48	<b>Inverse R-ch Gamma Corrected Value 243</b>
		Default Value: f300h Format: U16
	47:32	<b>Inverse Pixel Value 243</b>
		Default Value: f300h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 243</b>
		Default Value: f300h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 243</b>
		Default Value: f300h Format: U16
974..975	63:48	<b>Forward R-ch Gamma Corrected Value 243</b>
		Default Value: f300h Format: U16
	47:32	<b>Forward Pixel Value 243</b>
		Default Value: f300h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 243</b>
		Default Value: f300h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 243</b>
		Default Value: f300h Format: U16
976..977	63:48	<b>Inverse R-ch Gamma Corrected Value 244</b>
		Default Value: f400h Format: U16
	47:32	<b>Inverse Pixel Value 244</b>
		Default Value: f400h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 244</b>
		Default Value: f400h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 244</b>
		Default Value: f400h Format: U16
978..979	63:48	<b>Forward R-ch Gamma Corrected Value 244</b>
		Default Value: f400h Format: U16
	47:32	<b>Forward Pixel Value 244</b>
		Default Value: f400h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 244</b>
		Default Value: f400h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 244</b>
		Default Value: f400h Format: U16
980..981	63:48	<b>Inverse R-ch Gamma Corrected Value 245</b>
		Default Value: f500h Format: U16
	47:32	<b>Inverse Pixel Value 245</b>
		Default Value: f500h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 245</b>
		Default Value: f500h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 245</b>
		Default Value: f500h Format: U16
982..983	63:48	<b>Forward R-ch Gamma Corrected Value 245</b>
		Default Value: f500h Format: U16
	47:32	<b>Forward Pixel Value 245</b>
		Default Value: f500h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 245</b>
		Default Value: f500h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 245</b>
		Default Value: f500h Format: U16
984..985	63:48	<b>Inverse R-ch Gamma Corrected Value 246</b>
		Default Value: f600h Format: U16
	47:32	<b>Inverse Pixel Value 246</b>
		Default Value: f600h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 246</b>
		Default Value: f600h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 246</b>
		Default Value: f600h Format: U16
986..987	63:48	<b>Forward R-ch Gamma Corrected Value 246</b>
		Default Value: f600h Format: U16
	47:32	<b>Forward Pixel Value 246</b>
		Default Value: f600h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 246</b>
		Default Value: f600h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 246</b>
		Default Value: f600h Format: U16
988..989	63:48	<b>Inverse R-ch Gamma Corrected Value 247</b>
		Default Value: f700h Format: U16
	47:32	<b>Inverse Pixel Value 247</b>
		Default Value: f700h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 247</b>
		Default Value: f700h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 247</b>
		Default Value: f700h Format: U16
990..991	63:48	<b>Forward R-ch Gamma Corrected Value 247</b>
		Default Value: f700h Format: U16
	47:32	<b>Forward Pixel Value 247</b>
		Default Value: f700h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 247</b>
		Default Value: f700h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 247</b>
		Default Value: f700h Format: U16
992..993	63:48	<b>Inverse R-ch Gamma Corrected Value 248</b>
		Default Value: f800h Format: U16
	47:32	<b>Inverse Pixel Value 248</b>
		Default Value: f800h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 248</b>
		Default Value: f800h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 248</b>
		Default Value: f800h Format: U16
994..995	63:48	<b>Forward R-ch Gamma Corrected Value 248</b>
		Default Value: f800h Format: U16
	47:32	<b>Forward Pixel Value 248</b>
		Default Value: f800h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 248</b>
		Default Value: f800h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 248</b>
		Default Value: f800h Format: U16
996..997	63:48	<b>Inverse R-ch Gamma Corrected Value 249</b>
		Default Value: f900h Format: U16
	47:32	<b>Inverse Pixel Value 249</b>
		Default Value: f900h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 249</b>
		Default Value: f900h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 249</b>
		Default Value: f900h Format: U16
998..999	63:48	<b>Forward R-ch Gamma Corrected Value 249</b>
		Default Value: f900h Format: U16
	47:32	<b>Forward Pixel Value 249</b>
		Default Value: f900h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 249</b>
		Default Value: f900h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 249</b>
		Default Value: f900h Format: U16
1000..1001	63:48	<b>Inverse R-ch Gamma Corrected Value 250</b>
		Default Value: fa00h Format: U16
	47:32	<b>Inverse Pixel Value 250</b>
		Default Value: fa00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 250</b>
		Default Value: fa00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 250</b>
		Default Value: fa00h Format: U16
1002..1003	63:48	<b>Forward R-ch Gamma Corrected Value 250</b>
		Default Value: fa00h Format: U16
	47:32	<b>Forward Pixel Value 250</b>
		Default Value: fa00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 250</b>
		Default Value: fa00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 250</b>
		Default Value: fa00h Format: U16
1004..1005	63:48	<b>Inverse R-ch Gamma Corrected Value 251</b>
		Default Value: fb00h Format: U16
	47:32	<b>Inverse Pixel Value 251</b>
		Default Value: fb00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 251</b>
		Default Value: fb00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 251</b>
		Default Value: fb00h Format: U16
1006..1007	63:48	<b>Forward R-ch Gamma Corrected Value 251</b>
		Default Value: fb00h Format: U16
	47:32	<b>Forward Pixel Value 251</b>
		Default Value: fb00h Format: U16



<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 251</b>
		Default Value: fb00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 251</b>
		Default Value: fb00h Format: U16
1008..1009	63:48	<b>Inverse R-ch Gamma Corrected Value 252</b>
		Default Value: fc00h Format: U16
	47:32	<b>Inverse Pixel Value 252</b>
		Default Value: fc00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 252</b>
		Default Value: fc00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 252</b>
		Default Value: fc00h Format: U16
1010..1011	63:48	<b>Forward R-ch Gamma Corrected Value 252</b>
		Default Value: fc00h Format: U16
	47:32	<b>Forward Pixel Value 252</b>
		Default Value: fc00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 252</b>
		Default Value: fc00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 252</b>
		Default Value: fc00h Format: U16
1012..1013	63:48	<b>Inverse R-ch Gamma Corrected Value 253</b>
		Default Value: fd00h Format: U16
	47:32	<b>Inverse Pixel Value 253</b>
		Default Value: fd00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Inverse B-ch Gamma Corrected Value 253</b>
		Default Value: fd00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 253</b>
		Default Value: fd00h Format: U16
1014..1015	63:48	<b>Forward R-ch Gamma Corrected Value 253</b>
		Default Value: fd00h Format: U16
	47:32	<b>Forward Pixel Value 253</b>
		Default Value: fd00h Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 253</b>
		Default Value: fd00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 253</b>
		Default Value: fd00h Format: U16
1016..1017	63:48	<b>Inverse R-ch Gamma Corrected Value 254</b>
		Default Value: fe00h Format: U16
	47:32	<b>Inverse Pixel Value 254</b>
		Default Value: fe00h Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 254</b>
		Default Value: fe00h Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 254</b>
		Default Value: fe00h Format: U16
1018..1019	63:48	<b>Forward R-ch Gamma Corrected Value 254</b>
		Default Value: fe00h Format: U16
	47:32	<b>Forward Pixel Value 254</b>
		Default Value: fe00h Format: U16

<b>Gamut_Expansion_Gamma_Correction</b>		
	31:16	<b>Forward B-ch Gamma Corrected Value 254</b>
		Default Value: fe00h Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 254</b>
		Default Value: fe00h Format: U16
1020..1021	63:48	<b>Inverse R-ch Gamma Corrected Value 255</b>
		Default Value: ffffh Format: U16
	47:32	<b>Inverse Pixel Value 255</b>
		Default Value: ffffh Format: U16
	31:16	<b>Inverse B-ch Gamma Corrected Value 255</b>
		Default Value: ffffh Format: U16
	15:0	<b>Inverse G-ch Gamma Corrected Value 255</b>
		Default Value: ffffh Format: U16
1022..1023	63:48	<b>Forward R-ch Gamma Corrected Value 255</b>
		Default Value: ffffh Format: U16
	47:32	<b>Forward Pixel Value 255</b>
		Default Value: ffffh Format: U16
	31:16	<b>Forward B-ch Gamma Corrected Value 255</b>
		Default Value: ffffh Format: U16
	15:0	<b>Forward G-ch Gamma Corrected Value 255</b>
		Default Value: ffffh Format: U16



## GGTT Entry

GGTT_ENTRY - GGTT Entry			
Size (in bits):	64		
Default Value:	0x00000000, 0x00000000		
The GPU accesses GGTT table entries as uncacheable.			
DWord	Bit	Description	
0	63:54	<b>Reserved</b>	
	53	<b>Reserved</b>	
	52	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	51:46	<b>Reserved</b>	
	45:12	<b>Address</b>	
		Physical address of 4KB memory page referenced by this entry.	
	11:5	<b>Reserved</b>	
4:2	<b>Function Number</b>		
	The Number of the Function to which this page has been assigned. This field is ignored unless HW Graphics Virtualization (SRIOV) is enabled.		
1	<b>Ignored</b>		
	<i>Ignored (h/w does not care about values behind ignored registers)</i>		
0	<b>Present</b>		
	When set to 1, indicates that this Entry is Valid, and the corresponding page is Present in physical memory		

## GraphicsAddress63-0

<b>GA63_0 - GraphicsAddress63-0</b>		
Size (in bits):		64
Default Value:		0x00000000, 0x00000000
DWord	Bit	Description
0..1	63:57	<b>Reserved</b>
		Access: RO
		Format: MBZ
	56:48	<b>Reserved</b>
		Access: RO
		Format: MBZ
	47:0	<b>GraphicsAddress47-0</b>
		Format: GraphicsAddress[47:0]
	<p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and must be zero. For the specified product generations, use 48-bit addressing.</p>	



## GraphicsAddress63-1

<b>GA63_1 - GraphicsAddress63-1</b>		
Size (in bits):	63	
Default Value:	0x00000000, 0x00000000	
<p>This structure is intended to define the upper bits of the GraphicsAddress, when bit 0 is already defined in the referring register. So bit 0 of this structure should correspond to bit 1 of the full GraphicsAddress.</p>		
DWord	Bit	Description
0..1	62:56	<b>Reserved</b>
		Access: RO
		Format: MBZ
	55:47	<b>Reserved</b>
		Access: RO
		Format: MBZ
46:0	<b>GraphicsAddress47-1</b>	
	Format: GraphicsAddress[47:1]	
	<p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and must be zero.            Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] would be a 4KB page address.</p>	
	<p>Bits 47:1 of a 48-bit GraphicsAddress. Look for bit 0 definition in the referring register.</p>	

## GraphicsAddress63-2

<b>GA63_2 - GraphicsAddress63-2</b>		
Size (in bits):	62	
Default Value:	0x00000000, 0x00000000	
<p>This structure is intended to define the upper bits of the GraphicsAddress, when bits 1:0 are already defined in the referring register. So bit 0 of this structure should correspond to bit 2 of the full GraphicsAddress.</p>		
DWord	Bit	Description
0..1	61:55	<b>Reserved</b>
		Access: RO
		Format: MBZ
	54:46	<b>Reserved</b>
		Access: RO
		Format: MBZ
	45:0	<b>GraphicsAddress47-2</b>
		Format: GraphicsAddress[47:2]
		<p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and must be zero. Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] would be a 4KB page address.</p>
<p>Bits 47:2 of a 48-bit GraphicsAddress. Look for the definition of bits 1:0 in the referring register.</p>		

## GraphicsAddress63-3

<b>GA63_3 - GraphicsAddress63-3</b>		
Size (in bits):	61	
Default Value:	0x00000000, 0x00000000	
This structure is intended to define the upper bits of the GraphicsAddress, when bits 2:0 are already defined in the referring register. So bit 0 of this structure should correspond to bit 3 of the full GraphicsAddress.		
DWord	Bit	Description
0..1	60:54	<b>Reserved</b>
		Access: RO
		Format: MBZ
	53:45	<b>Reserved</b>
		Access: RO
		Format: MBZ
	44:0	<b>GraphicsAddress47-3</b>
		Format: GraphicsAddress[47:3]
		GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and must be zero. Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] would be a 4KB page address.
Bits 47:2 of a 48-bit GraphicsAddress. Look for the definition of bits 2:0 in the referring register.		



## GraphicsAddress63-4

<b>GA63_4 - GraphicsAddress63-4</b>		
Size (in bits):	60	
Default Value:	0x00000000, 0x00000000	
<p>This structure is intended to define the upper bits of the GraphicsAddress, when bits 3:0 are already defined in the referring register. So bit 0 of this structure should correspond to bit 4 of the full GraphicsAddress.</p>		
DWord	Bit	Description
0..1	59:53	<b>Reserved</b>
		Access: RO
		Format: MBZ
	52:44	<b>Reserved</b>
		Access: RO
		Format: MBZ
	43:0	<b>GraphicsAddress47-4</b>
		Format: GraphicsAddress[47:4]
		<p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and must be zero. Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] would be a 4KB page address.</p>
<p>Bits 47:4 of a 48-bit GraphicsAddress. Look for the definition of bits 3:0 in the referring register.</p>		



## GraphicsAddress63-5

<b>GA63_5 - GraphicsAddress63-5</b>		
Size (in bits):	59	
Default Value:	0x00000000, 0x00000000	
<p>This structure is intended to define the upper bits of the GraphicsAddress, when bits 4:0 are already defined in the referring register. So bit 0 of this structure should correspond to bit 5 of the full GraphicsAddress.</p>		
DWord	Bit	Description
0..1	58:52	<b>Reserved</b>
		Access: RO
		Format: MBZ
	51:43	<b>Reserved</b>
		Access: RO
		Format: MBZ
	42:0	<b>GraphicsAddress47-5</b>
		Format: GraphicsAddress[47:5]
		<p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and must be zero. Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] would be a 4KB page address.</p>
<p>Bits 47:5 of a 48-bit GraphicsAddress. Look for the definition of bits 4:0 in the referring register.</p>		

## GraphicsAddress63-6

<b>GA63_6 - GraphicsAddress63-6</b>		
Size (in bits):	58	
Default Value:	0x00000000, 0x00000000	
<p>This structure is intended to define the upper bits of the GraphicsAddress, when bits 5:0 are already defined in the referring register. So bit 0 of this structure should correspond to bit 6 of the full GraphicsAddress.</p>		
DWord	Bit	Description
0..1	57:51	<b>Reserved</b>
		Access: RO
		Format: MBZ
	50:42	<b>Reserved</b>
		Access: RO
		Format: MBZ
	41:0	<b>GraphicsAddress47-6</b>
		Format: GraphicsAddress[47:6]
		<p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and must be zero. Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] would be a 4KB page address.</p>
<p>Bits 47:6 of a 48-bit GraphicsAddress. Look for the definition of bits 5:0 in the referring register.</p>		



## GraphicsAddress63-12

<b>GA63_12 - GraphicsAddress63-12</b>		
Size (in bits):	52	
Default Value:	0x00000000, 0x00000000	
<p>This structure is intended to define the upper bits of the GraphicsAddress, when bits 11:0 are already defined in the referring register. So bit 0 of this structure should correspond to bit 12 of the full GraphicsAddress.</p>		
DWord	Bit	Description
0..1	51:45	<b>Reserved</b>
		Access: RO
		Format: MBZ
	44:36	<b>Reserved</b>
		Access: RO
		Format: MBZ
	35:0	<b>GraphicsAddress47-12</b>
		Format: GraphicsAddress[47:12]
		<p>GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and must be zero. Some GraphicsAddress fields only specify the upper address bits. For example GraphicsAddress[47:12] would be a 4KB page address.</p>
<p>Bits 47:12 of a 48-bit GraphicsAddress. Look for the definition of bits 11:0 in the referring register.</p>		

## GTC Interrupt Bit Definition

GTC Interrupt Bit Definition			
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	<b>GTC Lock Loss</b> GTC has lost lock with a remote GTC sink. The difference between the local and remote GTC has exceeded programmed threshold.	
	30:22	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	21	<b>GTC Aux Rx Error USBC6</b> An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.	
	20	<b>GTC Update Complete USBC6</b> A hardware initiated GTC update has completed with a sink attached to this port.	
	19	<b>GTC Aux Rx Error USBC5</b> An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.	
	18	<b>GTC Update Complete USBC5</b> A hardware initiated GTC update has completed with a sink attached to this port.	
	17	<b>GTC Aux Rx Error USBC4</b> An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.	
	16	<b>GTC Update Complete USBC4</b> A hardware initiated GTC update has completed with a sink attached to this port.	
	15	<b>GTC Aux Rx Error USBC3</b> An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.	
	14	<b>GTC Update Complete USBC3</b> A hardware initiated GTC update has completed with a sink attached to this port.	
	13	<b>GTC Aux Rx Error USBC2</b> An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.	
	12	<b>GTC Update Complete USBC2</b> A hardware initiated GTC update has completed with a sink attached to this port.	
	11	<b>GTC Aux Rx Error USBC1</b> An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.	
10	<b>GTC Update Complete USBC1</b> A hardware initiated GTC update has completed with a sink attached to this port.		
9:6	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	

## GTC Interrupt Bit Definition

5	<b>GTC Aux Rx Error DDIC</b> An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.
4	<b>GTC Update Complete DDIC</b> A hardware initiated GTC update has completed with a sink attached to this port.
3	<b>GTC Aux Rx Error DDIB</b> An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.
2	<b>GTC Update Complete DDIB</b> A hardware initiated GTC update has completed with a sink attached to this port.
1	<b>GTC Aux Rx Error DDIA</b> An aux channel error occurred during GTC transfer with remote GTC sink attached to this port.
0	<b>GTC Update Complete DDIA</b> A hardware initiated GTC update has completed with a sink attached to this port.

## GTPM Interrupt Vector

GTPM_INTR_VEC - GTPM Interrupt Vector			
Size (in bits):	16		
Default Value:	0x00000000		
DWord	Bit	Description	
0	15:14	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	13	<b>Unslice Frequency Control Up Interrupt</b> This interrupt is no longer used	
	12	<b>Unslice Frequency Control Down Interrupt</b> This interrupt is no longer used	
	11	<b>NFADFL Frequency Up Interrupt</b> This interrupt is no longer used	
	10	<b>NFADFL Frequency Down Interrupt</b> This interrupt is no longer used	
	9	<b>ARAT Interrupt</b> Always Running Apic Timer Interrupt. This interrupt is sent by GPM to GuCmicro-controller (for scheduling purposes). Host SW does not require this interrupt (so this interrupt is not sent to G-unit).	
	8	<b>GTPM Engines Idle Interrupt</b>	
	7	<b>GTPM Uncore to Core Trap Interrupt</b>	
	6	<b>GTPM Render Frequency Downwards Timeout During RC6 Interrupt</b> This interrupt is no longer used	
	5	<b>GTPM Render P-State Up Threshold Interrupt</b> This interrupt is no longer used	
	4	<b>GTPM Render P-State Down Threshold Interrupt</b> This interrupt is no longer used	
	3	<b>Spare 3</b> This interrupt is no longer used	
	2	<b>GTPM Render Geyserville Up Evaluation Interval Interrupt</b> This interrupt is no longer used	
	1	<b>GTPM Render Geyserville Down Evaluation Interval Interrupt</b> This interrupt is no longer used	
0		<b>Reserved</b>	
		Access:	RO
		Format:	MBZ

## GUC Interrupt Vector

<b>GUC_INTR_VEC - GUC Interrupt Vector</b>		
Size (in bits):	16	
Default Value:	0x00000000	
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0	15	<b>GUC Interrupt to Host</b>
	14	<b>GUC Execution Error</b>
	13	<b>GUC Display Event Received</b>
	12	<b>GUC Semaphore Signaled</b>
	11	<b>IOMMU Sent Message to GUC</b>
	10	<b>GUC Doorbell Rang</b>
	9	<b>GUC DMA Done</b>
	8	<b>GUC Fatal Error</b>
	7	<b>GUC Notification Error</b>
	6	<b>GUC SW interrupt 6</b>
	5	<b>GUC SW interrupt 5</b>
	4	<b>GUC SW interrupt 4</b>
	3	<b>GUC SW interrupt 3</b>
	2	<b>GUC SW interrupt 2</b>
	1	<b>GUC SW interrupt 1</b>
	0	<b>GUC SW interrupt 0</b>



## G-Unit Interrupt Vector

<b>GUNIT_INTR_VEC - G-Unit Interrupt Vector</b>			
Size (in bits):	16		
Default Value:	0x00000000		
This interrupt is delivered to GuC to indicate a request for Function Level Reset of a specific Virtual Function in Virtualization mode. GuC uses this request to clean up internal queues.			
<b>DWord</b>	<b>Bit</b>	<b>Description</b>	
0	15:8	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	7	<b>Function Level Reset Request For Virtual Function 7</b>	
	6	<b>Function Level Reset Request For Virtual Function 6</b>	
	5	<b>Function Level Reset Request For Virtual Function 5</b>	
	4	<b>Function Level Reset Request For Virtual Function 4</b>	
	3	<b>Function Level Reset Request For Virtual Function 3</b>	
	2	<b>Function Level Reset Request For Virtual Function 2</b>	
	1	<b>Function Level Reset Request For Virtual Function 1</b>	
0	<b>Function Level Reset Request For Virtual Function 0 (Physical Function)</b>		



## Half Precision Dual Source SIMD8 Message Data Payload Register

<b>MDPR_DSH_SIMD8 - Half Precision Dual Source SIMD8 Message Data Payload Register</b>				
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	<b>Src0 Data1</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 0 slot 1 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Src0 Data0</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 0 slot 0 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
1	31:16	<b>Src0 Data3</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 0 slot 3 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Src0 Data2</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 0 slot 2 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
2	31:16	<b>Src0 Data5</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 0 slot 5 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Src0 Data4</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 0 slot 4 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
3	31:16	<b>Src0 Data7</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 0 slot 7 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Src0 Data6</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 0 slot 6 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			

## MDPR\_DSH\_SIMD8 - Half Precision Dual Source SIMD8 Message Data Payload Register

4	31:16	<b>Src1 Data1</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 1 slot 1 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Src1 Data0</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 1 slot 0 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
5	31:16	<b>Src1 Data3</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 1 slot 3 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Src1 Data2</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 1 slot 2 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
6	31:16	<b>Src1 Data5</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 1 slot 5 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Src1 Data4</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 1 slot 4 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
7	31:16	<b>Src1 Data7</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 1 slot 7 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Src1 Data6</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the source 1 slot 6 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			



## Half Precision OM Replicated SIMD16 Render Target Data Payload

<b>MDP_RTWH_M16REP - Half Precision OM Replicated SIMD16 Render Target Data Payload</b>				
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	<b>oMask</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>MDPR_OMASK</b></td> </tr> </table> Slots [15:0] oMask	Format:	<b>MDPR_OMASK</b>
Format:	<b>MDPR_OMASK</b>			
1.0-1.7	255:0	<b>RGBA</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>MDPR_H_RGBA</b></td> </tr> </table> RGBA for all slots [15:0]	Format:	<b>MDPR_H_RGBA</b>
Format:	<b>MDPR_H_RGBA</b>			



## Half Precision OM S0A SIMD16 Render Target Data Payload

MDP_RTWH_MA16 - Half Precision OM S0A SIMD16 Render Target Data Payload		
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Source 0 Alpha[15:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_H_SIMD16</span> Slots [15:0] Source 0 Alpha
1.0-1.7	255:0	<b>oMask</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_OMASK</span> Slots [15:0] oMask
2.0-2.7	255:0	<b>Red[15:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_H_SIMD16</span> Slots [15:0] Red
3.0-3.7	255:0	<b>Green[15:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_H_SIMD16</span> Slots [15:0] Green
4.0-4.7	255:0	<b>Blue[15:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_H_SIMD16</span> Slots [15:0] Blue
5.0-5.7	255:0	<b>Alpha[15:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_H_SIMD16</span> Slots [15:0] Alpha

## Half Precision OM SIMD8 Dual Source Render Target Data Payload

<b>MDP_RTWH_M8DS - Half Precision OM SIMD8 Dual Source Render Target Data Payload</b>		
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000,	
DWord	Bit	Description
0.0-0.7	255:0	<b>oMask</b> Format: <b>MDPR_OMASK</b> oMask for slots [7:0] and [15:8]. Operation selects upper or lower half.
1.0-1.7	255:0	<b>Red</b> Format: <b>MDPR_DSH_SIMD8</b> Slots[7:0] or [15:8] of Src0 and Src1 Red
2.0-2.7	255:0	<b>Green</b> Format: <b>MDPR_DSH_SIMD8</b> Slots[7:0] or [15:8] of Src0 and Src1 Green
3.0-3.7	255:0	<b>Blue</b> Format: <b>MDPR_DSH_SIMD8</b> Slots[7:0] or [15:8] of Src0 and Src1 Blue
4.0-4.7	255:0	<b>Alpha</b> Format: <b>MDPR_DSH_SIMD8</b> Slots[7:0] or [15:8] of Src0 and Src1 Alpha









## MDP\_RTWH\_SMA8 - Half Precision OS OM S0A SIMD8 Render Target Data Payload

6.0-6.7	255:0	<b>Stencil</b>	
		Format:	<b>MDPR_STENCIL</b>
		Slots [7:0] Stencil	



## Half Precision OS OM SIMD8 Render Target Data Payload

MDP_RTWH_SM8 - Half Precision OS OM SIMD8 Render Target Data Payload		
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>oMask</b> Format: <b>MDPR_OMASK</b> Slots [7:0] oMask. Upper half ignored.
1.0-1.7	255:0	<b>Red</b> Format: <b>MDPR_H_SIMD8</b> Slots [7:0] Red
2.0-2.7	255:0	<b>Green</b> Format: <b>MDPR_H_SIMD8</b> Slots [7:0] Green
3.0-3.7	255:0	<b>Blue</b> Format: <b>MDPR_H_SIMD8</b> Slots [7:0] Blue
4.0-4.7	255:0	<b>Alpha</b> Format: <b>MDPR_H_SIMD8</b> Slots [7:0] Alpha
5.0-5.7	255:0	<b>Stencil</b> Format: <b>MDPR_STENCIL</b> 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		
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Source 0 Alpha</b> Format: <b>MDPR_H_SIMD8</b> Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	<b>Red</b> Format: <b>MDPR_H_SIMD8</b> Slots [7:0] Red
2.0-2.7	255:0	<b>Green</b> Format: <b>MDPR_H_SIMD8</b> Slots [7:0] Green
3.0-3.7	255:0	<b>Blue</b> Format: <b>MDPR_H_SIMD8</b> Slots [7:0] Blue
4.0-4.7	255:0	<b>Alpha</b> Format: <b>MDPR_H_SIMD8</b> Slots [7:0] Alpha
5.0-5.7	255:0	<b>Stencil</b> Format: <b>MDPR_STENCIL</b> Slots [7:0] Stencil



# Half Precision OS SIMD8 Dual Source Render Target Data Payload

MDP_RTWH_S8DS - Half Precision OS SIMD8 Dual Source Render Target Data Payload		
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Red</b> Format: <span style="border: 1px solid black; padding: 2px;"><b>MDPR_DSH_SIMD8</b></span> Slots[7:0] or [15:8] of Src0 and Src1 Red
1.0-1.7	255:0	<b>Green</b> Format: <span style="border: 1px solid black; padding: 2px;"><b>MDPR_DSH_SIMD8</b></span> Slots[7:0] or [15:8] of Src0 and Src1 Green
2.0-2.7	255:0	<b>Blue</b> Format: <span style="border: 1px solid black; padding: 2px;"><b>MDPR_DSH_SIMD8</b></span> Slots[7:0] or [15:8] of Src0 and Src1 Blue
3.0-3.7	255:0	<b>Alpha</b> Format: <span style="border: 1px solid black; padding: 2px;"><b>MDPR_DSH_SIMD8</b></span> Slots[7:0] or [15:8] of Src0 and Src1 Alpha
4.0-4.7	255:0	<b>Stencil</b> Format: <span style="border: 1px solid black; padding: 2px;"><b>MDPR_STENCIL</b></span> Slots [7:0] or [15:8] of Stencil









## MDP\_RTWH\_SZMA8 - Half Precision OS SZ OM S0A SIMD8 Render Target Data Payload

6.0-6.7	255:0	<b>Source Depth</b>	
		Format:	<b>MDP_DW_SIMD8</b>
		Slots [7:0] Source Depth	
7.0-7.7	255:0	<b>Stencil</b>	
		Format:	<b>MDPR_STENCIL</b>
		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	<b>Stencil</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_STENCIL</span> 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	<b>Stencil</b> Format: <b>MDPR_STENCIL</b> Slots [7:0] Stencil
---------	-------	--





## MDP\_RTWH\_SZA8 - Half Precision OS SZ S0A SIMD8 Render Target Data Payload

6.0-6.7	255:0	<b>Stencil</b>	
		Format:	<b>MDPR_STENCIL</b>
Slots [7:0] Stencil			







## Half Precision Replicated Pixel Render Target Data Payload Register

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



## Half Precision Replicated SIMD16 Render Target Data Payload

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









## Half Precision SIMD8 Message Data Payload Register

<b>MDPR_H_SIMD8 - Half Precision SIMD8 Message Data Payload Register</b>				
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	<b>Data1</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 1 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data0</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 0 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
1	31:16	<b>Data3</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 3 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data2</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 2 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
2	31:16	<b>Data5</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 5 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data4</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 4 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
3	31:16	<b>Data7</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 7 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data6</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 6 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			



## MDPR\_H\_SIMD8 - Half Precision SIMD8 Message Data Payload Register

4..7	127:0	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ



## Half Precision SIMD8 Render Target Data Payload

<b>MDP_RTWH_8 - Half Precision SIMD8 Render Target Data Payload</b>				
Size (in bits):	1024			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<b>Red</b> <table border="1"><tr><td>Format:</td><td><b>MDPR_H_SIMD8</b></td></tr></table> Slots [7:0] Red	Format:	<b>MDPR_H_SIMD8</b>
Format:	<b>MDPR_H_SIMD8</b>			
1.0-1.7	255:0	<b>Green</b> <table border="1"><tr><td>Format:</td><td><b>MDPR_H_SIMD8</b></td></tr></table> Slots [7:0] Green	Format:	<b>MDPR_H_SIMD8</b>
Format:	<b>MDPR_H_SIMD8</b>			
2.0-2.7	255:0	<b>Blue</b> <table border="1"><tr><td>Format:</td><td><b>MDPR_H_SIMD8</b></td></tr></table> Slots [7:0] Blue	Format:	<b>MDPR_H_SIMD8</b>
Format:	<b>MDPR_H_SIMD8</b>			
3.0-3.7	255:0	<b>Alpha</b> <table border="1"><tr><td>Format:</td><td><b>MDPR_H_SIMD8</b></td></tr></table> Slots [7:0] Alpha	Format:	<b>MDPR_H_SIMD8</b>
Format:	<b>MDPR_H_SIMD8</b>			

## Half Precision SIMD16 Message Data Payload Register

<b>MDPR_H_SIMD16 - Half Precision SIMD16 Message Data Payload Register</b>				
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	<b>Data1</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 1 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data0</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 0 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
1	31:16	<b>Data3</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 3 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data2</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 2 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
2	31:16	<b>Data5</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 5 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data4</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 4 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
3	31:16	<b>Data7</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 7 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data6</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 6 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			

## MDPR\_H\_SIMD16 - Half Precision SIMD16 Message Data Payload Register

4	31:16	<b>Data9</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 9 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data8</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 8 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
5	31:16	<b>Data11</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 11 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data10</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 10 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
6	31:16	<b>Data13</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 13 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data12</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 12 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			
7	31:16	<b>Data15</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 15 data in this payload register	Format:	IEEE_FLOAT16
	Format:	IEEE_FLOAT16		
15:0	<b>Data14</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT16</td> </tr> </table> Specifies the slot 14 data in this payload register	Format:	IEEE_FLOAT16	
Format:	IEEE_FLOAT16			

## Half Precision SIMD16 Render Target Data Payload

MDP_RTWH_16 - Half Precision SIMD16 Render Target Data Payload				
Size (in bits):	1024			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<b>Red[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MDPR_H_SIMD16</td> </tr> </table> Slots [15:0] Red	Format:	MDPR_H_SIMD16
Format:	MDPR_H_SIMD16			
1.0-1.7	255:0	<b>Green[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MDPR_H_SIMD16</td> </tr> </table> Slots [15:0] Green	Format:	MDPR_H_SIMD16
Format:	MDPR_H_SIMD16			
2.0-2.7	255:0	<b>Blue[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MDPR_H_SIMD16</td> </tr> </table> Slots [15:0] Blue	Format:	MDPR_H_SIMD16
Format:	MDPR_H_SIMD16			
3.0-3.7	255:0	<b>Alpha[15:0]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>MDPR_H_SIMD16</td> </tr> </table> Slots [15:0] Alpha	Format:	MDPR_H_SIMD16
Format:	MDPR_H_SIMD16			



## MDP\_RTWH\_ZMA8 - Half Precision SZ OM S0A SIMD8 Render Target Data Payload

6.0-6.7	255:0	<b>Source Depth</b>	
		Format:	<b>MDP_DW_SIMD8</b>
		Slots [7:0] Source Depth	





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



## Half Precision SZ OM SIMD8 Render Target Data Payload

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



## Half Precision SZ OM SIMD16 Render Target Data Payload

MDP_RTWH_ZM16 - Half Precision SZ OM SIMD16 Render Target Data Payload		
Size (in bits):	1792	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>oMask</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_OMASK</span> Slots [15:0] oMask
1.0-1.7	255:0	<b>Red[15:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_H_SIMD16</span> Slots [15:0] Red
2.0-2.7	255:0	<b>Green[15:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_H_SIMD16</span> Slots [15:0] Green
3.0-3.7	255:0	<b>Blue[15:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_H_SIMD16</span> Slots [15:0] Blue
4.0-4.7	255:0	<b>Alpha[15:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_H_SIMD16</span> Slots [15:0] Alpha
5.0-5.7	255:0	<b>Source Depth[7:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Source Depth

MDP_RTWH_ZM16 - Half Precision SZ OM SIMD16 Render Target Data Payload		
6.0-6.7	255:0	<b>Source Depth[15:8]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [15:8] Source Depth



## Half Precision SZ S0A SIMD16 Render Target Data Payload

MDP_RTWH_ZA16 - Half Precision SZ S0A SIMD16 Render Target Data Payload		
Size (in bits):	1792	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Source 0 Alpha[15:0]</b> Format: MDP_H_SIMD16 Slots [15:0] Source 0 Alpha
1.0-1.7	255:0	<b>Red[15:0]</b> Format: MDP_H_SIMD16 Slots [15:0] Red
2.0-2.7	255:0	<b>Green[15:0]</b> Format: MDP_H_SIMD16 Slots [15:0] Green
3.0-3.7	255:0	<b>Blue[15:0]</b> Format: MDP_H_SIMD16 Slots [15:0] Blue
4.0-4.7	255:0	<b>Alpha[15:0]</b> Format: MDP_H_SIMD16 Slots [15:0] Alpha
5.0-5.7	255:0	<b>Source Depth[7:0]</b> Format: MDP_DW_SIMD8 Slots [7:0] Source Depth



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



## Half Precision SZ SIMD8 Dual Source Render Target Data Payload

<b>MDP_RTWH_Z8DS - Half Precision SZ SIMD8 Dual Source Render Target Data Payload</b>		
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Red</b> Format: <b>MDPR_DSH_SIMD8</b> Slots[7:0] or [15:8] of Src0 and Src1 Red
1.0-1.7	255:0	<b>Green</b> Format: <b>MDPR_DSH_SIMD8</b> Slots[7:0] or [15:8] of Src0 and Src1 Green
2.0-2.7	255:0	<b>Blue</b> Format: <b>MDPR_DSH_SIMD8</b> Slots[7:0] or [15:8] of Src0 and Src1 Blue
3.0-3.7	255:0	<b>Alpha</b> Format: <b>MDPR_DSH_SIMD8</b> Slots[7:0] or [15:8] of Src0 and Src1 Alpha
4.0-4.7	255:0	<b>Source Depth</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] or [15:8] of Source Depth



## Half Precision SZ SIMD16 Render Target Data Payload

MDP_RTWH_Z16 - Half Precision SZ SIMD16 Render Target Data Payload		
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Red[15:0]</b> Format: <b>MDPR_H_SIMD16</b> Slots [15:0] Red
1.0-1.7	255:0	<b>Green[15:0]</b> Format: <b>MDPR_H_SIMD16</b> Slots [15:0] Green
2.0-2.7	255:0	<b>Blue[15:0]</b> Format: <b>MDPR_H_SIMD16</b> Slots [15:0] Blue
3.0-3.7	255:0	<b>Alpha[15:0]</b> Format: <b>MDPR_H_SIMD16</b> Slots [15:0] Alpha
4.0-4.7	255:0	<b>Source Depth[7:0]</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source Depth
5.0-5.7	255:0	<b>Source Depth[15:8]</b> Format: <b>MDP_DW_SIMD8</b> 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	31:8	<b>Reserved</b>					
		Access: RO					
		Format: MBZ					
	7	<b>Reserved</b>					
	6:3	<b>Reserved</b>					
		Access: RO					
		Format: MBZ					
	2	<b>Command Privilege Violation Error</b> 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	<b>Reserved</b>					
		Access: RO					
		Format: MBZ					
	0	<b>Instruction Error</b> 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"> <li>Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported).</li> <li>Defeatured MI Instruction Opcodes:</li> </ul> <table border="1"> <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> <p style="text-align: center;"><b>Programming Notes</b></p> This error indications cannot be cleared except by reset (i.e., it is a fatal error).	Value	Name	Description	1	
Value	Name	Description					
1		Instruction Error detected					











<b>Hardware Status Page Layout</b>						
4	31:0	<b>Ring Head Pointer Storage</b> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">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).</td> </tr> </tbody> </table>	Description		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).	
Description						
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	351:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					
16..39	767:0	<b>Context Status DWords</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td><b>CONTEXT STATUS[12]</b></td> </tr> </table>	Format:	<b>CONTEXT STATUS[12]</b>		
Format:	<b>CONTEXT STATUS[12]</b>					
40..46 These dwords are reserved.	223:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					
47	31:0	<b>Last Written Status Offset</b>				
48..1023	31231:0	<b>General Purpose</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 55%;">Format:</td> <td>U32[976]</td> </tr> </table> <p>These locations can be used for general purpose via the MI_STORE_DATA_INDEX or MI_STORE_DATA_IMM instructions.</p>	Format:	U32[976]		
Format:	U32[976]					



## 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	<p><b>Indirect Payload Data Size in bits</b></p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Number of bits to be inserted. Not including those skipped bytes in the beginning. For VP9: the Data is always valid from start of cache-line, no offset is allowed.</p>	Format:	U32				
Format:	U32							
1..2	63:0	<p><b>Indirect Payload Base Address</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>SplitBaseAddress64ByteAligned</b></td> </tr> </table> <p>48-bit address of the indirect payload data in memory buffer.</p> <table border="1"> <tr> <th colspan="2">Programming Notes</th> </tr> <tr> <td colspan="2">Payload must begin in a byte position, but the payload can be ended in a bit position.</td> </tr> </table>	Format:	<b>SplitBaseAddress64ByteAligned</b>	Programming Notes		Payload must begin in a byte position, but the payload can be ended in a bit position.	
Format:	<b>SplitBaseAddress64ByteAligned</b>							
Programming Notes								
Payload must begin in a byte position, but the payload can be ended in a bit position.								
3	31:0	<p><b>Indirect Payload Base Address</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>MemoryAddressAttributes</b></td> </tr> </table>	Format:	<b>MemoryAddressAttributes</b>				
Format:	<b>MemoryAddressAttributes</b>							

## HCP\_REF\_LIST\_ENTRY

HCP_REF_LIST_ENTRY			
Size (in bits):	32		
Default Value:	0x00000000		
DWord	Bit	Description	
0	31:16	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	15	<b>bottom_field_flag</b>	
		Format:	U1
		Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.	
		Value	Name
		0	Bottom Field
		1	Top Field
		<b>Programming Notes</b>	
	Not supported in encoder mode.		
	14	<b>field_pic_flag</b>	
Format:		U1	
Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.			
Value		Name	
0		Video Frame	
1		Video Field	
<b>Programming Notes</b>			
Not supported in encoder mode.			
13	<b>LongTermReference</b>		
	Format:	U1	
	Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.		
	Value	Name	
	0	Short term reference	
1	Long term reference		

## HCP\_REF\_LIST\_ENTRY

12	<b>luma_weight_IX_flag</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U1</td> </tr> </table> <p>Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Default weighted prediction for luma</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Explicit weighted prediction for Luma</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	Default weighted prediction for luma	1	Explicit weighted prediction for Luma
Format:	U1									
Value	Name									
0	Default weighted prediction for luma									
1	Explicit weighted prediction for Luma									
11	<b>chroma_weight_IX_flag</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U1</td> </tr> </table> <p>Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Default weighted prediction for Chroma</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Explicit weighted prediction for Chroma</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	Default weighted prediction for Chroma	1	Explicit weighted prediction for Chroma
Format:	U1									
Value	Name									
0	Default weighted prediction for Chroma									
1	Explicit weighted prediction for Chroma									
10:8	<b>list_entry_IX: Reference Picture Frame ID (RefAddr[0-7])</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U3</td> </tr> </table> <p>Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.</p> <p>The reference picture frame ID identifies the reference picture associated with the base address defined in <b>Reference Picture Address (RefAddr[0-7])</b> in the HCP_PIPE_BUF_ADDR_STATE command.</p>	Format:	U3						
Format:	U3									
7:0	<b>Reference Picture tb Value</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U8</td> </tr> </table> <p>Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15.</p> <p>clip(-128,127, CurrentPOC - RefPOC), where RefPOC is the POC value of the reference picture. 8-bit signed.</p> <p>See the "Derivation process for temporal luma motion vector prediction" in the HEVC standard.</p>	Format:	U8						
Format:	U8									

## HCP\_TILE\_POSITION\_IN\_CTB

HCP_TILE_POSITION_IN_CTB		
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:24	<b>CtbPos3+i</b> Format: <input type="text"/> U8
	23:16	<b>CtbPos2+i</b> Format: <input type="text"/> U8
	15:8	<b>CtbPos1+i</b> Format: <input type="text"/> U8
	7:0	<b>CtbPos0+i</b> Format: <input type="text"/> U8



## HCP\_TILE\_POSITION\_IN\_CTB\_MSB

HCP_TILE_POSITION_IN_CTB_MSB			
Size (in bits):	64		
Default Value:	0x00000000, 0x00000000		
Added to support 16k picture size.			
DWord	Bit	Description	
0..1	63:44	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	43:42	<b>Ctb position of tile 21 [9:8]</b> MSB 2 bits of CTB row position of tile row 21.	
		<b>Programming Notes</b>	
		Please note that this field is MBZ for columns	
	41:40	<b>Ctb row position of tile column 20 [9:8]</b> MSB 2 bits of CTB row position of tile row 20.	
		<b>Programming Notes</b>	
		Please note that this field is MBZ for columns	
	39:38	<b>Ctb row position of tile column 19 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 19.	
	37:36	<b>Ctb row position of tile column 18 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 18.	
	35:34	<b>Ctb row position of tile column 17 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 17.	
	33:32	<b>Ctb row position of tile column 16 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 16.	
	31:30	<b>Ctb row position of tile column 15 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 15.	
29:28	<b>Ctb row position of tile column 14 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 14.		
27:26	<b>Ctb row position of tile column 13 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 13.		
25:24	<b>Ctb row position of tile column 12 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 12.		
23:22	<b>Ctb row position of tile column 11 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 11.		
21:20	<b>Ctb row position of tile column 10 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 10.		
19:18	<b>Ctb row position of tile column 9 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 9.		

<b>HCP_TILE_POSITION_IN_CTB_MSB</b>	
17:16	<b>Ctb row position of tile column 8 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 8.
15:14	<b>Ctb row position of tile column 7 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 7.
13:12	<b>Ctb row position of tile column 6 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 6.
11:10	<b>Ctb row position of tile column 5 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 5.
9:8	<b>Ctb row position of tile column 4 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 4.
7:6	<b>Ctb row position of tile column 3 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 3.
5:4	<b>Ctb row position of tile column 2 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 2.
3:2	<b>Ctb row position of tile column 1 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 1.
1:0	<b>Ctb row position of tile column 0 [9:8]</b> MSB 2 bits of CTB row or column position of tile row or column 0.

## HCP\_WEIGHTOFFSET\_CHROMA\_ENTRY

HCP_WEIGHTOFFSET_CHROMA_ENTRY		
Source:	VideoCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:24	<b>ChromaOffsetLX [i][1]</b>
		Where X is the RefPicListNum and i is the list entry number 0 through 15. DW 18 corresponds to i=0, DW 33 corresponds to i=15.
		Valid only if explicit weighted prediction for chroma is enabled, otherwise must be zero.
		<b>Programming Notes</b>
		This (combined with its MSbyte below) shall be in the range of -WpOffsetHalfRangeC to (WpOffsetHalfRangeC - 1), inclusive WpOffsetHalfRangeC = $1 \ll (\text{high\_precision\_offsets\_enabled\_flag} ? (\text{BitDepthC} - 1) : 7)$
23:16		<b>delta_chroma_weight_IX[i][1]</b>
		Format: <span style="float: right;">S7</span>
		Where X is the RefPicListNum and i is the list entry number 0 through 15. DW 18 corresponds to i=0, DW 33 corresponds to i=15.
		Valid only if explicit weighted prediction for chroma is enabled, otherwise must be zero.
		<b>Programming Notes</b>
		This shall be in the range of 128 to 127, inclusive
15:8		<b>ChromaOffsetLX[i][0]</b>
		Where X is the RefPicListNum and i is the list entry number 0 through 15. DW 18 corresponds to i=0, DW 33 corresponds to i=15.
		Valid only if explicit weighted prediction for chroma is enabled, otherwise must be zero.
		<b>Programming Notes</b>
		This (combined with its MSbyte below) shall be in the range of -WpOffsetHalfRangeC to (WpOffsetHalfRangeC - 1), inclusive WpOffsetHalfRangeC = $1 \ll (\text{high\_precision\_offsets\_enabled\_flag} ? (\text{BitDepthC} - 1) : 7)$
7:0		<b>delta_chroma_weight_IX[i][0]</b>
		Format: <span style="float: right;">S7</span>
		Where X is the RefPicListNum and i is the list entry number 0 through 15. DW 18 corresponds to i=0, DW 33 corresponds to i=15.
		Valid only if explicit weighted prediction for chroma is enabled, otherwise must be zero.



<b>HCP_WEIGHTOFFSET_CHROMA_ENTRY</b>		
		<b>Programming Notes</b>
		This shall be in the range of 128 to 127, inclusive

## HCP\_WEIGHTOFFSET\_CHROMA\_EXT\_ENTRY

HCP_WEIGHTOFFSET_CHROMA_EXT_ENTRY		
Source:	VideoCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:24	<p><b>ChromaOffsetLX[i+1][1] MSByte</b>            To support 4:4:4, the chroma offset is extended into 16-bit.            In order to keep SW back compatible, the most significant byte is programmed here.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This is only MSByte portion of ChromaOffsetLX. Please refer to LSB section for available range.</p>
	23:16	<p><b>ChromaOffsetLX[i][1] MSByte</b>            To support 4:4:4, the chroma offset is extended into 16-bit.            In order to keep SW back compatible, the most significant byte is programmed here.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This is only MSByte portion of ChromaOffsetLX. Please refer to LSB section for available range.</p>
	15:8	<p><b>ChromaOffsetLX[i+1][0] MSByte</b>            To support 4:4:4, the chroma offset is extended into 16-bit.            In order to keep SW back compatible, the most significant byte is programmed here.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This is only MSByte portion of ChromaOffsetLX. Please refer to LSB section for available range.</p>
	7:0	<p><b>ChromaOffsetLX[i][0] MSByte</b>            To support 4:4:4, the chroma offset is extended into 16-bit.            In order to keep SW back compatible, the most significant byte is programmed here.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This is only MSByte portion of ChromaOffsetLX. Please refer to LSB section for available range.</p>

## HCP\_WEIGHTOFFSET\_LUMA\_ENTRY

HCP_WEIGHTOFFSET_LUMA_ENTRY				
Source:	VideoCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:24	<b>luma_offset_IX[i] MSByte</b> To support 4:4:4, the luma offset is extended into 16-bit. In order to keep SW back compatible, the most significant byte is programmed here.		
		<p style="text-align: center;"><b>Programming Notes</b></p> This is only MSByte portion of luma_offset_IX. Please refer to LSB section for available range.		
	23:16	<b>Reserved</b>		
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
15:8	15:8	<b>luma_offset_IX[i]</b> Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15. Valid only if explicit weighted prediction for luma is enabled, otherwise must be zero.		
		<p style="text-align: center;"><b>Programming Notes</b></p> This (combined with its MSbyte above) shall be in the range of $WpOffsetHalfRange_v$ to $WpOffsetHalfRange_v + 1$ , where $WpOffsetHalfRange_v = 1 \ll (high\_precision\_offsets\_enabled\_flag ? (BitDepth_v - 1) : 7)$		
	7:0	<b>delta_luma_weight_IX[i]</b>		
		<table border="1"> <tr> <td>Format:</td> <td>S7</td> </tr> </table>	Format:	S7
Format:	S7			
		Where X is the RefPicListNum and i is the list entry number 0 through 15. DW2 corresponds to i=0, DW17 corresponds to i=15. Valid only if explicit weighted prediction for luma is enabled, otherwise must be zero.		
		<p style="text-align: center;"><b>Programming Notes</b></p> When luma_weight_l0_flag[i] is equal to 1, the value of delta_luma_weight_l0[i] shall be in the range of 128 to 127, inclusive.		

## Header

<b>Header</b>		
Size (in bits):	16	
Default Value:	0x00000000	
DWord	Bit	Description
0	15:12	<b>SWSB[7:4]</b> Format: <span style="float: right;"><b>SWSB[7:4]</b></span> This field specify the Software Scoreboard information.
	11:8	<b>SWSB[3:0]</b> Format: <span style="float: right;"><b>SWSB[3:0]</b></span> This field specify the Software Scoreboard information.
	7	<b>Reserved</b> Format: <span style="float: right;">MBZ</span>
	6:0	<b>Opcode</b> Format: <span style="float: right;"><b>EU_OPCODE</b></span> This field determines the operation performed by the instruction.

## Header Forbidden Message Descriptor Control Field

<b>MDC_MHF - Header Forbidden Message Descriptor Control Field</b>											
Size (in bits):	1										
Default Value:	0x00000000										
DWord	Bit	Description									
0	0	<p><b>Message Header Present</b> Indicates the message forbids a 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 <b>[Default]</b></td> <td>Message header is not present</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td>Not used</td> </tr> </tbody> </table>	Value	Name	Description	0h	No <b>[Default]</b>	Message header is not present	1h	Reserved	Not used
Value	Name	Description									
0h	No <b>[Default]</b>	Message header is not present									
1h	Reserved	Not used									

## Header Present Message Descriptor Control Field

MDC_MHP - Header Present Message Descriptor Control Field													
Size (in bits):	1												
Default Value:	0x00000000												
DWord	Bit	Description											
0	0	<p><b>Message Header Present</b></p> <table border="1"> <tr> <td>Format:</td> <td>Boolean</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:	Boolean	Value	Name	Description	0h	No	Message header is not present	1h	Yes	Message header is present
Format:	Boolean												
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																	
Size (in bits):		1															
Default Value:		0x00000001															
DWord	Bit	Description															
0	0	<p><b>Message Header Present</b></p> <table border="1"> <tr> <td colspan="2">Format:</td> <td>Boolean</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 <b>[Default]</b></td> <td>Message header is present</td> </tr> </table>	Format:		Boolean	Indicates the message requires a message header.			Value	Name	Description	0h	Reserved	Not used	1h	Yes <b>[Default]</b>	Message header is present
Format:		Boolean															
Indicates the message requires a message header.																	
Value	Name	Description															
0h	Reserved	Not used															
1h	Yes <b>[Default]</b>	Message header is present															



## HEVC\_ARBITRATION\_PRIORITY

HEVC_ARBITRATION_PRIORITY												
Size (in bits):	2											
Default Value:	0x00000000											
This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.												
DWord	Bit	Description										
0	1:0	<b>Priority</b>										
		Format: U2										
		<table border="1"><thead><tr><th>Value</th><th>Name</th></tr></thead><tbody><tr><td>00b</td><td>Highest priority</td></tr><tr><td>01b</td><td>Second highest priority</td></tr><tr><td>10b</td><td>Third highest priority</td></tr><tr><td>11b</td><td>Lowest priority</td></tr></tbody></table>	Value	Name	00b	Highest priority	01b	Second highest priority	10b	Third highest priority	11b	Lowest priority
Value	Name											
00b	Highest priority											
01b	Second highest priority											
10b	Third highest priority											
11b	Lowest priority											



## HEVC\_VP9\_RDOQ\_LAMBDA\_FIELDS

HEVC_VP9_RDOQ_LAMBDA_FIELDS		
Source:	VideoCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:16	<b>LambdaValue1</b> Lambda value for Intra/Inter Luma/Chroma component of QP=1, 3, , 61, 63...73,75 (odd number) For 12-bit video, the QP range has extended to include 65 to 75. for HEVC
	15:0	<b>LambdaValue0</b> Lambda value for Intra/Inter Luma/Chroma component of QP=0, 2, , 60, 62,...72,74(even number) For 12-bit video, the QP range has extended to include 64to 74. for HEVC



## HUC\_VIRTUAL\_ADDR\_REGION

HUC_VIRTUAL_ADDR_REGION		
Source:	VideoCS	
Size (in bits):	96	
Default Value:	0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..2	95:64	<b>HUC Surface (VirtualAddrRegion[0-15])</b> Format: <b>MemoryAddressAttributes</b>
	63:0	<b>HUC Surface Base Address (VirtualAddrRegion[0-15])</b> Format: <b>SplitBaseAddress4KByteAligned</b> HUC surface base address for each virtual address region [0-15].

## HW Generated BINDING\_TABLE\_STATE

HW Generated BINDING_TABLE_STATE				
Size (in bits):	16			
Default Value:	0x00000000			
<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	<b>Surface State Pointer</b> <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

<b>MDP_HW1 - Hword 1 Block Data Payload</b>		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0.0-0.7	255:0	<b>Hword</b> Format: U256 Specifies the Hword data

## Hword 2 Block Data Payload

<b>MDP_HW2 - Hword 2 Block Data Payload</b>		
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	<b>Hword0</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U256</td> </tr> </table> <p>Specifies the Hword data for element 0</p>
Format:	U256	
1.0-1.7	255:0	<b>Hword1</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%;">U256</td> </tr> </table> <p>Specifies the Hword data for element 1</p>
Format:	U256	



## Hword 4 Block Data Payload

<b>MDP_HW4 - Hword 4 Block Data Payload</b>				
Size (in bits):	1024			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<b>Hword0</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U256</td> </tr> </table> Specifies the Hword data for element 0	Format:	U256
Format:	U256			
1.0-1.7	255:0	<b>Hword1</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U256</td> </tr> </table> Specifies the Hword data for element 1	Format:	U256
Format:	U256			
2.0-2.7	255:0	<b>Hword2</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U256</td> </tr> </table> Specifies the Hword data for element 2	Format:	U256
Format:	U256			
3.0-3.7	255:0	<b>Hword3</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U256</td> </tr> </table> Specifies the Hword data for element 3	Format:	U256
Format:	U256			





MDP_HW8 - Hword 8 Block Data Payload		
6.0-6.7	255:0	<b>Hword6</b>
		Format: U256 Specifies the Hword data for element 6
7.0-7.7	255:0	<b>Hword7</b>
		Format: U256 Specifies the Hword data for element 7



## Hword Channel Mode Message Header Control

<b>MHC_A64_CMODE - Hword Channel Mode Message Header Control</b>		
Size (in bits):		32
Default Value:		0x00000000
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0	31:0	<b>Reserved</b>
		Access: RO
		Format: MBZ



## Hword Register Blocks Message Descriptor Control Field

<b>MDC_DB_HW - Hword Register Blocks Message Descriptor Control Field</b>																	
Size (in bits):		2															
Default Value:		0x00000000															
DWord	Bit	Description															
0	1:0	<b>Register Blocks</b> Specifies the number of Hword blocks to be read or written															
		<table border="1"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>00h</td><td>HW1</td><td>1 Hword register</td></tr><tr><td>01h</td><td>HW2</td><td>2 Hword registers</td></tr><tr><td>02h</td><td>HW4</td><td>4 Hword registers</td></tr><tr><td>03h</td><td>HW8</td><td>8 Hword registers</td></tr></tbody></table>	Value	Name	Description	00h	HW1	1 Hword register	01h	HW2	2 Hword registers	02h	HW4	4 Hword registers	03h	HW8	8 Hword registers
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

<b>MH_IGNORE - Ignored Message Header</b>						
Source:	EuSubFunctionDataPort0					
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	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					



## IndirectOperand

IndirectOperand			
Size (in bits):		14	
Default Value:		0x00000000	
DWord	Bit	Description	
0	13:10	<b>AddrSubRegNum</b> Format: <table border="1"><tr><td>AddrSubRegNum</td></tr></table>	AddrSubRegNum
	AddrSubRegNum		
9:0	<b>AddrImm</b> Format: <table border="1"><tr><td>S9</td></tr></table> <p>This field defines a 10-bit signed integer offset in units of byte, only used with the Indirect Addressing Mode. In that addressing mode, the Address Immediate Offset value is added to an address subregister value to determine the operand's address in the GRF.</p>	S9	
S9			

## 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	<b>Concealment Method</b> 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	<b>Init Current MB Number</b> 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	<b>Intra PredMode (4x4/8x8 Luma) Error Control Bit</b> 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.	<b>MB Error Concealment B Temporal Prediction mode</b> 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><b>[Default]</b></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	<b>[Default]</b>	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	<b>[Default]</b>	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	<b>Reserved</b> Access: RO Format: MBZ														

## Inline Data Description for MFD\_AVC\_BSD\_Object

25	<b>MB Error Concealment B Temporal Motion Vectors Override Enable Flag</b>	<p>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.</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 style="text-align: center;">0</td> <td style="text-align: center;">[Default]</td> <td>Predicted Motion Vectors are used during MB Concealment</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>Motion Vectors are Overridden to 0 during MB Concealment</td> </tr> </tbody> </table>	Value	Name	Description	0	[Default]	Predicted Motion Vectors are used during MB Concealment	1		Motion Vectors are Overridden to 0 during MB Concealment			
Value	Name	Description												
0	[Default]	Predicted Motion Vectors are used during MB Concealment												
1		Motion Vectors are Overridden to 0 during MB Concealment												
24	<b>MB Error Concealment B Temporal Weight Prediction Disable Flag</b>	<p>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.</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 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												
23:22	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
Access:	RO													
Format:	MBZ													
21:16	<b>Concealment Picture ID</b>	<p>This field identifies the picture in the reference list to be used for concealment. This field is only valid if <b>Concealment Method</b> is Inter P Copy.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Bit Filed</th> <th style="width: 10%;">Value</th> <th style="width: 80%;">Defenition</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">21</td> <td style="text-align: center;">0</td> <td>Frame Picture</td> </tr> <tr> <td style="text-align: center;">21</td> <td style="text-align: center;">1</td> <td>Field picture</td> </tr> <tr> <td style="text-align: center;">20:16</td> <td style="text-align: center;">All</td> <td>Frame Store Index[4:0]</td> </tr> </tbody> </table>	Bit Filed	Value	Defenition	21	0	Frame Picture	21	1	Field picture	20:16	All	Frame Store Index[4:0]
Bit Filed	Value	Defenition												
21	0	Frame Picture												
21	1	Field picture												
20:16	All	Frame Store Index[4:0]												
15	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
Access:	RO													
Format:	MBZ													
14	<b>BSD Premature Complete Error Handling</b>	<p>BSD Premature Complete Error occurs in situation where the Slice decode is completed but there are still data in the bitstream.</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 style="text-align: center;">1</td> <td></td> <td>Set the interrupt to the driver (provide MMIO registers for MB address R/W)</td> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>Ignore the error and continue (masked the interrupt), assume the hardware automatically performs the error handling</td> </tr> </tbody> </table>	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			
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	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
Access:	RO													
Format:	MBZ													

## Inline Data Description for MFD\_AVC\_BSD\_Object

12	<b>MPR Error (MV out of range) Handling</b> Software must follow the action for each Value as follow:	
	<b>Value</b>	<b>Name</b>
		<b>Description</b>
	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	<b>Reserved</b> Access: RO Format: MBZ	
10	<b>Entropy Error Handling</b> Software must follow the action for each Value as follow:	
	<b>Value</b>	<b>Name</b>
		<b>Description</b>
	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	<b>Reserved</b> Access: RO Format: MBZ	
8	<b>MB Header Error Handling</b> Software must follow the action for each Value as follow:	
	<b>Value</b>	<b>Name</b>
		<b>Description</b>
	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	<b>MB Error Concealment B Spatial Prediction mode</b> These two bits control how the reference L0/L1 are overridden in B spatial slice.	
	<b>Value</b>	<b>Name</b>
		<b>Description</b>
	00b	<b>[Default]</b> 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	<b>Reserved</b> Access: RO Format: MBZ	

## Inline Data Description for MFD\_AVC\_BSD\_Object

4		<b>MB Error Concealment B Spatial Motion Vectors Override Disable Flag</b>	<p>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.</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 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										
3		<b>MB Error Concealment B Spatial Weight Prediction Disable Flag</b>	<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 style="width: 75%;">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		<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="width: 40%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO											
Format:	MBZ											
1		<b>MB Error Concealment P Slice Motion Vectors Override Disable Flag</b>	<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 style="width: 75%;">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		<b>MB Error Concealment P Slice Weight Prediction Disable Flag</b>	<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 style="width: 75%;">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.										
1	31:16	<b>First MB Byte Offset of Slice Data or Slice Header</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 100%; text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">MFX supports only DXVA2 Long and Short Format.</td> </tr> </tbody> </table>	Programming Notes	MFX supports only DXVA2 Long and Short Format.							
Programming Notes												
MFX supports only DXVA2 Long and Short Format.												
	15:8	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="width: 40%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO											
Format:	MBZ											



## Inline Data Description for MFD\_AVC\_BSD\_Object

	7	<b>Fix Prev Mb Skipped</b> 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.										
	6:5	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ					
Access:	RO											
Format:	MBZ											
	4	<b>Emulation Prevention Byte Present</b> <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;">0</td> <td></td> <td>H/W needs to perform Emulation Byte Removal</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>H/W does not need to perform Emulation Byte Removal</td> </tr> </tbody> </table>		Value	Name	Description	0		H/W needs to perform Emulation Byte Removal	1		H/W does not need to perform Emulation Byte Removal
Value	Name	Description										
0		H/W needs to perform Emulation Byte Removal										
1		H/W does not need to perform Emulation Byte Removal										
	3	<b>LastSlice Flag</b> 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. <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;">1</td> <td></td> <td>If the current Slice to be decoded is the very last slice of the current picture.</td> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>If the current Slice to be decoded is any slice other than the very last slice of the current picture</td> </tr> </tbody> </table>		Value	Name	Description	1		If the current Slice to be decoded is the very last slice of the current picture.	0		If the current Slice to be decoded is any slice other than the very last slice of the current picture
Value	Name	Description										
1		If the current Slice to be decoded is the very last slice of the current picture.										
0		If the current Slice to be decoded is any slice other than the very last slice of the current picture										
	2:0	<b>First Macroblock (MB)Bit Offset</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Exists If:</td> <td>//AVC Long Format Only</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>This field provides the bit offset of the first macroblock of the Slice in the first byte of the input compressed bitstream.</p>		Exists If:	//AVC Long Format Only	Format:	U3					
Exists If:	//AVC Long Format Only											
Format:	U3											
2	31	<b>I Slice Concealment Mode</b> This field controls how AVC decoder handle MB concealment in I Slice <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="width: 35%;">Value</th> <th style="width: 65%;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>Intra Concealment</td> </tr> <tr> <td style="text-align: center;">0</td> <td>Inter Concealment</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #e1eef6;"> <th style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">           If this field is set to "0" (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.         </td> </tr> </tbody> </table>		Value	Name	1	Intra Concealment	0	Inter Concealment	Programming Notes	If this field is set to "0" (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.	
Value	Name											
1	Intra Concealment											
0	Inter Concealment											
Programming Notes												
If this field is set to "0" (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	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ					
Access:	RO											
Format:	MBZ											

## Inline Data Description for MFD\_AVC\_BSD\_Object

29:24	<b>Concealment Reference Picture + Field Bit</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U6</td> </tr> </table> <p>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</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Bit Filed</th> <th style="width: 15%;">Value</th> <th style="width: 70%;">Defenition</th> </tr> </thead> <tbody> <tr> <td>29</td> <td>MBZ</td> <td>is reserved for future expansion</td> </tr> <tr> <td>28:25</td> <td>All</td> <td>Reference PICTURE Number</td> </tr> <tr> <td>24</td> <td>All</td> <td>Field Bit(if the current picture is a field picture [Frame picture must be 0])</td> </tr> </tbody> </table>	Format:	U6	Bit Filed	Value	Defenition	29	MBZ	is reserved for future expansion	28:25	All	Reference PICTURE Number	24	All	Field Bit(if the current picture is a field picture [Frame picture must be 0])							
Format:	U6																						
Bit Filed	Value	Defenition																					
29	MBZ	is reserved for future expansion																					
28:25	All	Reference PICTURE Number																					
24	All	Field Bit(if the current picture is a field picture [Frame picture must be 0])																					
23	<b>P Slice Concealment Mode</b>	<p>This field controls how AVC decoder handle MB concealment in P Slice</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>1</td> <td>Intra Concealment</td> </tr> <tr> <td>0</td> <td>Inter Concealment</td> </tr> </tbody> </table>	Value	Name	1	Intra Concealment	0	Inter Concealment															
Value	Name																						
1	Intra Concealment																						
0	Inter Concealment																						
22:19	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td style="width: 40%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																	
Access:	RO																						
Format:	MBZ																						
18:16	<b>P Slice Inter Concealment Mode</b>	<p>This field controls how AVC decoder select reference picture for Concealment in P Slice.</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>000b</td> <td></td> <td>Top of Reference List L0 (Use top entry of Reference List L0)</td> </tr> <tr> <td>001b</td> <td></td> <td>Driver Specified Concealment Reference</td> </tr> <tr> <td>010b</td> <td></td> <td>Predicted Reference (Use reference picture predicted using P-Skip Algorithm)</td> </tr> <tr> <td>011b</td> <td></td> <td>Temporal Closest (Using POC to select the closest forward picture)[For L0: Closest POC smaller than current POC]</td> </tr> <tr> <td>100b</td> <td></td> <td>First Long Term Picture in Reference List L0 (If no long term picture available, use Temporal Closest Picture)</td> </tr> <tr> <td>101b-111b</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	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	
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>B Slice Concealment Mode</b>	<p>This field controls how AVC decoder handle MB concealment in B Slice</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>1</td> <td>Intra Concealment</td> </tr> <tr> <td>0</td> <td>Inter Concealment</td> </tr> </tbody> </table>	Value	Name	1	Intra Concealment	0	Inter Concealment															
Value	Name																						
1	Intra Concealment																						
0	Inter Concealment																						

## Inline Data Description for MFD\_AVC\_BSD\_Object

14	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
13:12	<b>B Slice Inter Direct Type Concealment Mode</b>	
	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.	
	<b>Value</b>	<b>Name</b> <b>Description</b>
	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	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
10:8	<b>B Slice Spatial Inter Concealment Mode</b>	
	This field controls how AVC decoder select reference picture for Spatial Inter Concealment in B Slice.	
	<b>Value</b>	<b>Name</b> <b>Description</b>
	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	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
6:4	<b>B Slice Temporal Inter Concealment Mode</b>	
	This field controls how AVC decoder select reference picture for Temporal Inter Concealment in B Slice	
	<b>Value</b>	<b>Name</b> <b>Description</b>
	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:	

## Inline Data Description for MFD\_AVC\_BSD\_Object

			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	<b>Reserved</b>		
	Access:		RO
	Format:		MBZ
1	<b>Intra 8x8/4x4 Prediction Error Concealment Control Bit</b> 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.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	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	<b>Intra Prediction Error Control Bit (applied to Intra16x16/Intra8x8/Intra4x4 Luma and Chroma)</b> This field controls if AVC decoder will fix Intra Prediction Mode if the decoded value is incorrect according to MB position.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	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 in MPEG2-IT Mode

Inline Data Description in MPEG2-IT Mode																						
Source:	VideoCS																					
Size (in bits):	192																					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000																					
<p>The content in this command is similar to that in the MEDIA_OBJECT command in IS mode described in the Media Chapter.</p> <p>Each MFD_IT_OBJECT command corresponds to the processing of one macroblock. Macroblock parameters are passed in as inline data and the non-zero DCT coefficient data for the macroblock is passed in as indirect data. Inline data starts at dword 7 of MFD_IT_OBJECT command. There are 7 dwords total.</p>																						
DWord	Bit	Description																				
0	31:28	<b>Motion Vertical Field Select</b> A bit-wise representation of a long [2][2] array as defined in #167;6.3.17.2 of the ISO/IEC 13818-2 (see also #167;7.6.4).																				
		<table border="1"> <thead> <tr> <th>Bit</th> <th>MVector[r]</th> <th>MVector[s]</th> <th>MotionVerticalFieldSelect Index</th> </tr> </thead> <tbody> <tr> <td>28</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>29</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>30</td> <td>1</td> <td>0</td> <td>2</td> </tr> <tr> <td>31</td> <td>1</td> <td>1</td> <td>3</td> </tr> </tbody> </table>	Bit	MVector[r]	MVector[s]	MotionVerticalFieldSelect Index	28	0	0	0	29	0	1	1	30	1	0	2	31	1	1	3
		Bit	MVector[r]	MVector[s]	MotionVerticalFieldSelect Index																	
		28	0	0	0																	
29	0	1	1																			
30	1	0	2																			
31	1	1	3																			
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Top Field</td> <td>The prediction is taken from the top reference field.</td> </tr> <tr> <td>1</td> <td>Bottom Field</td> <td>The prediction is taken from the bottom reference field.</td> </tr> </tbody> </table>	Value	Name	Description	0	Top Field	The prediction is taken from the top reference field.	1	Bottom Field	The prediction is taken from the bottom reference field.													
Value	Name	Description																				
0	Top Field	The prediction is taken from the top reference field.																				
1	Bottom Field	The prediction is taken from the bottom reference field.																				
27:26		<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																
Access:	RO																					
Format:	MBZ																					
25:24		<b>Motion Type</b> When combined with the destination picture type (field or frame) this Motion Type field indicates the type of motion to be applied to the macroblock. See ISO/IEC 13818-2 #167;6.3.17.1, Tables 6-17, 6-18. In particular, the device supports dual-prime motion prediction (11) in both frame and field picture type.																				
		<table border="1"> <thead> <tr> <th>Value</th> <th>Destination = Frame Picture_Structure = 11</th> <th>Destination = Field Picture_Structure != 11</th> </tr> </thead> <tbody> <tr> <td>'00'</td> <td>Reserved</td> <td>Reserved</td> </tr> <tr> <td>'01'</td> <td>Field</td> <td>Field</td> </tr> <tr> <td>'10'</td> <td>Frame</td> <td>16x8</td> </tr> <tr> <td>'11'</td> <td>Dual-Prime</td> <td>Dual-Prime</td> </tr> </tbody> </table>	Value	Destination = Frame Picture_Structure = 11	Destination = Field Picture_Structure != 11	'00'	Reserved	Reserved	'01'	Field	Field	'10'	Frame	16x8	'11'	Dual-Prime	Dual-Prime					
		Value	Destination = Frame Picture_Structure = 11	Destination = Field Picture_Structure != 11																		
		'00'	Reserved	Reserved																		
'01'	Field	Field																				
'10'	Frame	16x8																				
'11'	Dual-Prime	Dual-Prime																				

## Inline Data Description in MPEG2-IT Mode

23:22	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
21	<b>DCT Type</b>	
	This field specifies the DCT type of the current macroblock. The kernel should ignore this field when processing Cb/Cr data. See ISO/IEC 13818-2 #167;6.3.17.1. This field is zero if Coded Block Pattern is also zero (no coded blocks present).	
	<b>Value</b>	<b>Name</b>
	0	MC_FRAME_DCT
	1	MC_FIELD_DCT
		<b>Description</b>
		Macroblock is frame DCT coded
		Macroblock is field DCT coded
20:19	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
18	<b>Macroblock Motion Backward</b>	
	This field specifies if the backward motion vector is active. See ISO/IEC 13818-2 Tables B-2 through B-4.	
	<b>Value</b>	<b>Name</b>
	0	No backward motion vector
	1	Use backward motion vector(s)
17	<b>Macroblock Motion Forward</b>	
	This field specifies if the forward motion vector is active. See ISO/IEC 13818-2 Tables B-2 through B-4.	
	<b>Value</b>	<b>Name</b>
	0	No forward motion vector
	1	Use forward motion vector(s)
16	<b>Macroblock Intra Type</b>	
	This field specifies if the current macroblock is intra-coded. When set, Coded Block Pattern is ignored and no prediction is performed (i.e., no motion vectors are used). See ISO/IEC 13818-2 Tables B-2 through B-4.	
	<b>Value</b>	<b>Name</b>
	0	Non-intra macroblock
	1	Intra macroblock
15:12	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ

<b>Inline Data Description in MPEG2-IT Mode</b>					
	11:6 <b>Coded Block Pattern</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable[6]</td> </tr> </table> Bit 11: Y0 Bit 10: Y1 Bit 9: Y2 Bit 8: Y3 Bit 7: Cb4 Bit 6: Cr5	Format:	Enable[6]		
	Format:	Enable[6]			
	5:4 <b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Access:</td> <td style="width: 50%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO			
Format:	MBZ				
3 <b>LastMBInRow</b> This field indicates the last MB in each row					
2:0 <b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Access:</td> <td style="width: 50%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO				
Format:	MBZ				
1	31:16 <b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Access:</td> <td style="width: 50%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO			
	Format:	MBZ			
15:8 <b>VertOrigin</b> Vertical Origin In unit of macroblocks relative to the current picture (frame or field).					
7:0 <b>HorzOrigin</b> Horizontal Origin in unit of macroblocks.					
2	31:16 <b>Motion Vectors - Field 0, Forward, Vertical Component</b> Each vector component is a 16-bit two's-complement value. The vector is relative to the current macroblock location. According to ISO/IEC 13818-2 Table 7-8, the valid range of each vector component is [-2048, +2047.5], implying a format of s11.1. However, it should be noted that motion vector values are sign extended to 16 bits.				
	15:0 <b>Motion Vectors - Field 0, Forward, Horizontal Component</b>				
3	31:16 <b>Motion Vectors - Field 0, Backward, Vertical Component</b>				
	15:0 <b>Motion Vectors - Field 0, Backward, Horizontal Component</b>				
4	31:16 <b>Motion Vectors - Field 1, Forward, Vertical Component</b>				
	15:0 <b>Motion Vectors - Field 1, Forward, Horizontal Component</b>				
5	31:16 <b>Motion Vectors - Field 1, Backward, Vertical Component</b>				
	15:0 <b>Motion Vectors - Field 1, Backward, Horizontal Component</b>				

## 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	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	22:20	<b>MV Format(Motion Vector Size)</b>		
		Exists If:	//IntraMbFlag = 0	
		This field specifies the size and format of the output motion vectors.		
		<b>Value</b>	<b>Name</b>	<b>Description</b>
		000b	Intra MB	No Motion vectors
		100b	Inter Predict MB (Unpacked Motion Vector Mode)	Sixteen Motion Vectors Per MacroBlock
		Others	Reserved	
<b>Programming Notes</b>				
This field MBZ, when the <b>IntraMbFlag</b> = 1.				
19:18	<b>SegmentID</b>			
	Format:	U2		
Segment number 0-3				
17	<b>Enable Coeff Clamp</b>			
	<b>Value</b>	<b>Name</b>	<b>Description</b>	
	1		Magnitude of coefficients of the current MB is clamped based on the clamping matrix after quantization	
0		No Clamping		
16:14	<b>Reserved</b>			
	Access:	RO		
	Format:	MBZ		
13	<b>Intra MB Flag</b>			
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.				



## Inline Data Description - VP8 PAK OBJECT

		Value	Name	
		0h	INTER (Inter MacroBlock)	
		1h	INTRA (Intra MacroBlock)	
		<b>Programming Notes</b>		
		For I-picture MB (Intra MB Flag =1), this field must be set to 1.		
12:11	<b>RefPicSelect</b>	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.		
		Value	Name	
		00b	Last Frame	
		01b	Golden Frame	
		10b	Alt Frame	
10:8	<b>MB Type 3-Bits - Inter/Intra MB</b>	MB Type 3Bits [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		
		Value	Name	Description
		000b	16x16	<b>Inter MB</b> Only DW 6 bits 3:0 are used to indicate MVMode, MVMode can't be split
		001b	2 16x8 (mv_Top Bottom)	<b>Inter MB [10:8]</b> 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)	<b>Inter MB [10:8]</b> 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)	<b>Inter MB [10:8]</b> 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)	<b>Inter MB [10:8]</b> Split MV is inferred. There are 16 partitions. Each Sub-block uses 4 bits in DW6 and DW7.
		0b	16x16	<b>Intra MB [8]</b> 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	<b>Intra MB [8]</b> All bits in DW5 and DW6 are used to represent B_PRED modes (Bmodes) in each sub-blocks.
7:6	<b>Reserved</b>			
		Access:	RO	
		Format:	MBZ	

## Inline Data Description - VP8 PAK OBJECT

	5:4	<b>MB UV Mode</b>	
		<b>Value</b>	<b>Name</b>
		0	DC_PRED
		1	V_PRED
		2	H_PRED
	3	TM_PRED	
	3	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	2	<b>Skip MB Flag</b>	
		This field is equivalent to mb_skip_flag in VP8 spec.	
		<b>Programming Notes</b>	
		By setting this field to 1, it forces an Inter MacroBlock to be encoded as a skipped MacroBlock	
	1:0	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
1	31:24	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	23:16	<b>MbYCnt (Vertical Origin)</b>	
		Format:	U8
		This field specifies the vertical origin of current macroblock in the destination picture in units of macroblocks.	
	15:8	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	7:0	<b>MbXCnt (Horizontal Origin)</b>	
		Format:	U8
		This field specifies the horizontal origin of current macroblock in the destination picture in units of macroblocks.	
2	31:28	<b>B Mode for SubBlock7 (Y mode for the macroblock in non-B mode)</b>	
		For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.	
	27:24	<b>B Mode for SubBlock6 (Y mode for the macroblock in non-B mode)</b>	
		For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.	
	23:20	<b>B Mode for SubBlock5 (Y mode for the macroblock in non-B mode)</b>	
		For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.	

<b>Inline Data Description - VP8 PAK OBJECT</b>					
	19:16 <b>B Mode for SubBlock4 (Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	15:12 <b>B Mode for SubBlock3 (Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	11:8 <b>B Mode for SubBlock2 (Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	7:4 <b>B Mode for SubBlock1 (Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	3:0 <b>B Mode for SubBlock0 (Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
3	31:28 <b>B Mode for SubBlock15 (Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	27:24 <b>B Mode for SubBlock14(Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	23:20 <b>B Mode for SubBlock13(Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	19:16 <b>B Mode for SubBlock12(Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	15:12 <b>B Mode for SubBlock11(Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	11:8 <b>B Mode for SubBlock10 (Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	7:4 <b>B Mode for SubBlock9 (Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
	3:0 <b>B Mode for SubBlock8 (Y mode for the macroblock in non-B mode)</b> For Y-Mode and B-Mode Assignments refer to the assignment lists below this table.				
4	31:30 <b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO			
	Format:	MBZ			
29:16 <b>MV Y FWD 0</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>S13</td> </tr> </table> <p>The value of the y component of this motion vector for FWD block 0. Max value +/-1024 full pel (+/- 8192 1/8th pel) precision</p>	Format:	S13			
Format:	S13				
15:14 <b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO				
Format:	MBZ				

## Inline Data Description - VP8 PAK OBJECT

	13:0	<b>MV X FWD 0</b>	Format: S13	The value of the x component of this motion vector for FWD block 0. Max value +/-1024 full pel (+/- 8192 1/8th pel) precision
5	31:30	<b>Reserved</b>	Access: RO Format: MBZ	
	29:16	<b>MV Y FWD 1</b>	Format: S13	The value of the y component of this motion vector for FWD block 1.
	15:14	<b>Reserved</b>	Access: RO Format: MBZ	
	13:0	<b>MV X FWD 1</b>	Format: S13	The value of the x component of this motion vector for FWD block 1.
6	31:30	<b>Reserved</b>	Access: RO Format: MBZ	
	29:16	<b>MV Y FWD 2</b>	Format: S13	The value of the y component of this motion vector for FWD block 2.
	15:14	<b>Reserved</b>	Access: RO Format: MBZ	
	13:0	<b>MV X FWD 2</b>	Format: S13	The value of the x component of this motion vector for FWD block 2.
7	31:30	<b>Reserved</b>	Access: RO Format: MBZ	
	29:16	<b>MV Y FWD 3</b>	Format: S13	The value of the y component of this motion vector for FWD block 3.

Inline Data Description - VP8 PAK OBJECT					
	15:14	<b>Reserved</b>			
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				
	13:0	<b>MV X FWD 3</b>			
		<table border="1"> <tr> <td>Format:</td> <td>S13</td> </tr> </table> <p>The value of the x component of this motion vector for FWD block 3.</p>	Format:	S13	
Format:	S13				
8	31:30	<b>Reserved</b>			
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
29:16	<b>MV Y BWD 0</b>				
	<table border="1"> <tr> <td>Format:</td> <td>S13</td> </tr> </table> <p>The value of the y component of this motion vector for BWD block 0.</p>	Format:	S13		
Format:	S13				
15:14	<b>Reserved</b>				
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				
13:0	<b>MV X BWD 0</b>				
	<table border="1"> <tr> <td>Format:</td> <td>S13</td> </tr> </table> <p>The value of the x component of this motion vector for BWD block 0.</p>	Format:	S13		
Format:	S13				
9	31:30	<b>Reserved</b>			
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
29:16	<b>MV Y BWD 1</b>				
	<table border="1"> <tr> <td>Format:</td> <td>S13</td> </tr> </table> <p>The value of the y component of this motion vector for BWD block 1.</p>	Format:	S13		
Format:	S13				
15:14	<b>Reserved</b>				
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				
13:0	<b>MV X BWD 1</b>				
	<table border="1"> <tr> <td>Format:</td> <td>S13</td> </tr> </table> <p>The value of the x component of this motion vector for BWD block 1.</p>	Format:	S13		
Format:	S13				
10	31:30	<b>Reserved</b>			
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				

<b>Inline Data Description - VP8 PAK OBJECT</b>				
	29:16	<b>MV Y BWD 2</b>	Format: S13	The value of the y component of this motion vector for BWD block 2.
	15:14	<b>Reserved</b>	Access: RO Format: MBZ	
	13:0	<b>MV X BWD 2</b>	Format: S13	The value of the x component of this motion vector for BWD block 2.
11	31:30	<b>Reserved</b>	Access: RO Format: MBZ	
	29:16	<b>MV Y BWD 3</b>	Format: S13	The value of the y component of this motion vector for BWD block 3.
	15:14	<b>Reserved</b>	Access: RO Format: MBZ	
	13:0	<b>MV X BWD 3</b>	Format: S13	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><b>Kernel Start Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[31:6]</td> </tr> </table> <p>Specifies the 64-byte aligned address offset of the first instruction in the kernel. This pointer is relative to the <b>Instruction Base Address</b>.</p>	Format:	InstructionBaseOffset[31:6]						
	Format:	InstructionBaseOffset[31:6]								
5:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO									
Format:	MBZ									
1	31:16	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
	Access:	RO								
Format:	MBZ									
15:0	<p><b>Kernel Start Pointer High</b></p> <table border="1"> <tr> <td>Format:</td> <td>InstructionBaseOffset[47:32]</td> </tr> </table> <p>This field specifies the high 16 bits of starting address of the Kernel Pointer.</p>	Format:	InstructionBaseOffset[47:32]							
Format:	InstructionBaseOffset[47:32]									
2	31:21	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
	Access:	RO								
Format:	MBZ									
20	<p><b>Thread Preemption disable</b></p> <p>This field specifies whether, when dispatched, the thread is allowed to stop in middle on receiving mid-thread pre-emption request.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable <b>[Default]</b></td> <td>Thread is pre-empted on receiving pre-emption indication.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Thread is preempted only in case of page-fault.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable <b>[Default]</b>	Thread is pre-empted on receiving pre-emption indication.	1h	Enable	Thread is preempted only in case of page-fault.
Value	Name	Description								
0h	Disable <b>[Default]</b>	Thread is pre-empted on receiving pre-emption indication.								
1h	Enable	Thread is preempted only in case of page-fault.								
19	<p><b>Denorm Mode</b></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.								

## INTERFACE\_DESCRIPTOR\_DATA

18	<b>Single Program Flow</b>		
	Specifies whether the kernel program has a single program flow (SIMDn <sub>xm</sub> with m = 1) or multiple program flows (SIMDn <sub>xm</sub> with m > 1).		
	<b>Value</b>	<b>Name</b>	
	0h	Multiple	
	1h	Single	
	17	<b>Thread Priority</b>	
		Specifies the priority of the thread for dispatch.	
		<b>Value</b>	<b>Name</b>
		0h	Normal Priority
	1h	High Priority	
	16	<b>Floating Point Mode</b>	
		Specifies the floating point mode used by the dispatched thread.	
		<b>Value</b>	<b>Name</b>
		0h	IEEE-754
	1h	Alternate	
	15:14	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
13	<b>Illegal Opcode Exception Enable</b>		
	Format:	Enable	
This bit gets loaded into EU CR0.1[12] (note the bit # difference). See <i>Exceptions and ISA Execution Environment</i> .			
12	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
11	<b>Mask Stack Exception Enable</b>		
	Format:	Enable	
This bit gets loaded into EU CR0.1[11]. See <i>Exceptions and ISA Execution Environment</i> .			
10:8	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
7	<b>Software Exception Enable</b>		
	Format:	Enable	
This bit gets loaded into EU CR0.1[13] (note the bit # difference). See <i>Exceptions and ISA Execution Environment</i> .			



<b>INTERFACE_DESCRIPTOR_DATA</b>		
	6:0	<b>Reserved</b>
	Access:	RO
	Format:	MBZ
3	31:5	<b>Sampler State Pointer</b>
	Format:	DynamicStateOffset[31:5]SAMPLER_STATE
	Specifies the 32-byte aligned address offset of the sampler state table. This pointer is relative to the <b>Dynamic State Base Address</b> . <i>This field is ignored for child threads.</i>	
4	4:2	<b>Sampler Count</b>
	Format:	U3
	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>	
	<b>Value</b>	<b>Name</b>
	[0,4]	
	0h	No samplers used
	1h	Between 1 and 4 samplers used
	2h	Between 5 and 8 samplers used
	3h	Between 9 and 12 samplers used
	4h	Between 13 and 16 samplers used
	1:0	<b>Reserved</b>
	Access:	RO
	Format:	MBZ
4	31:16	<b>Reserved</b>
	Access:	RO
	Format:	MBZ
	15:5	<b>Binding Table Pointer</b>
	Format:	SurfaceStateOffset[15:5]SW Generated BINDING_TABLE_STATE*256 When HW Binding Table Alignment is alignment is clear.
Format:	SurfaceStateOffset[16:6]SW Generated BINDING_TABLE_STATE*256 [] When Binding Table Pool is Enabled.	
Format:	SurfaceStateOffset[18:8]SW Generated BINDING_TABLE_STATE*256 When HW Binding Table Alignment is alignment set to 512KB size	
<b>Description</b>		
Specifies a pointer offset into the binding table. <i>This field is ignored for child threads.</i>		
<ul style="list-style-type: none"> <li>When Binding Table Pool is Disabled, this pointer is relative to the <b>Surface State Base Address</b>.</li> </ul>		

<b>INTERFACE_DESCRIPTOR_DATA</b>																						
	<ul style="list-style-type: none"> <li>When Binding Table Pool is Enabled, this pointer is relative to the <b>Binding Table PoolBase Address</b>.</li> </ul>																					
4:0	<p><b>Binding Table Entry Count</b></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;"><b>Programming Notes</b></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	<p>31:16 <b>Constant/Indirect URB Entry Read Length</b></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 <b>Cross-Thread Constant Data Read Length</b> 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> <tr> <td>&gt;0</td> <td>=0</td> <td>Per-thread payload only</td> </tr> <tr> <td>&gt;0</td> <td>&gt;0</td> <td>Both kinds of payload</td> </tr> <tr> <td>=0</td> <td>&gt;0</td> <td>Only for CURBE payloads</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,63]</td> <td></td> </tr> </tbody> </table>	Format:	U16	Constant/Indirect URB Entry Read Length	Cross-Thread Constant Data Read Length	Notes	=0	=0	No Payload	>0	=0	Per-thread payload only	>0	>0	Both kinds of payload	=0	>0	Only for CURBE payloads	Value	Name	[0,63]	
Format:	U16																					
Constant/Indirect URB Entry Read Length	Cross-Thread Constant Data Read Length	Notes																				
=0	=0	No Payload																				
>0	=0	Per-thread payload only																				
>0	>0	Both kinds of payload																				
=0	>0	Only for CURBE payloads																				
Value	Name																					
[0,63]																						

<b>INTERFACE_DESCRIPTOR_DATA</b>																			
	15:0	<b>Constant URB Entry Read Offset</b> 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>Value</th> <th>Name</th> <th>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	<b>Reserved</b> Access: RO Format: MBZ																	
		<b>Rounding Mode</b> Format: U2 <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>RTNE <b>[Default]</b></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 <b>[Default]</b>	Round to Nearest Even	01b	RU	Round toward +Infinity	10b	RD	Round toward -Infinity	11b	RTZ	Round toward Zero		
	Value	Name	Description																
	00b	RTNE <b>[Default]</b>	Round to Nearest Even																
	01b	RU	Round toward +Infinity																
10b	RD	Round toward -Infinity																	
11b	RTZ	Round toward Zero																	
21	<b>Barrier Enable</b> Format: Enable This field specifies whether the thread group requires a barrier. If not, it can be dispatched without allocating one.																		
20:16	<b>Shared Local Memory Size</b> 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. uses a different encoding to allow encodings for the new 1k and 2k SLM sizes.																		
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Encodes 0K</td> <td>No SLM used</td> </tr> <tr> <td>1</td> <td>Encodes 1K</td> <td></td> </tr> <tr> <td>2</td> <td>Encodes 2K</td> <td></td> </tr> <tr> <td>3</td> <td>Encodes 4K</td> <td></td> </tr> <tr> <td>4</td> <td>Encodes 8K</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	
	Value	Name	Description																
0	Encodes 0K	No SLM used																	
1	Encodes 1K																		
2	Encodes 2K																		
3	Encodes 4K																		
4	Encodes 8K																		

INTERFACE_DESCRIPTOR_DATA			
		5	Encodes 16K
		6	Encodes 32K
		7	Encodes 64K
	15	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	14:13	<b>Over Dispatch Control</b>	
		Format:	U2
		Over Dispatch Control that enables and controls the amount of GPGPU thread over dispatch.	
		<b>Value</b>	<b>Name</b>
		00b	None <b>[Default]</b>
		11b	Normal
		01b	Low
		10b	High
		<b>Restriction</b>	
		Set this over dispatch control to None if Mid Thread Preemption is used.	
	12:10	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	9:0	<b>Number of Threads in GPGPU Thread Group</b>	
		Format:	U10
		Specifies the number of threads that are in this thread group.	
		<b>Value</b>	<b>Name</b>
		[1,112]	The minimum value is 1, while the maximum value is the number of threads in two subslices for local barriers. See vol1b Configurations for the number of threads per subslice for different products.
7	31:8	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	7:0	<b>Cross-Thread Constant Data Read Length</b>	
		Format:	U8
		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 <b>Constant URB Entry Read Length</b> .	
		<b>Value</b>	<b>Name</b>
		[0,127]	

# INTERRUPT

<b>INTERRUPT</b>												
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><b>ISR</b></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;"><b>Restriction</b></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><b>IMR</b></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 <b>[Default]</b></td> </tr> <tr> <td>0b</td> <td>Not Masked</td> </tr> <tr> <td>1b</td> <td>Masked</td> </tr> </tbody> </table>	Access:	R/W	Value	Name	FFFFFFFFh	All interrupts masked <b>[Default]</b>	0b	Not Masked	1b	Masked
Access:	R/W											
Value	Name											
FFFFFFFFh	All interrupts masked <b>[Default]</b>											
0b	Not Masked											
1b	Masked											
2	31:0	<p><b>IIR</b></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> <table border="1"> <thead> <tr> <th>Value</th> <th>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>	Access:	R/WC	Value	Name	0b	Condition Not Detected	1b	Condition Detected		
Access:	R/WC											
Value	Name											
0b	Condition Not Detected											
1b	Condition Detected											

<b>INTERRUPT</b>																			
	<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> </thead> <tbody> <tr> <td colspan="2">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.</td> </tr> </tbody> </table>	<b>Programming Notes</b>		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.															
<b>Programming Notes</b>																			
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.																			
3	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">31:0</td> <td colspan="2"><b>IER</b></td> </tr> <tr> <td>Access:</td> <td colspan="2" style="text-align: right;">R/W</td> </tr> <tr> <td colspan="3">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> <td style="text-align: center;"><b>Value</b></td> <td colspan="2" style="text-align: center;"><b>Name</b></td> </tr> <tr> <td style="text-align: center;">0b</td> <td colspan="2" style="text-align: center;">Disabled</td> </tr> <tr> <td style="text-align: center;">1b</td> <td colspan="2" style="text-align: center;">Enabled</td> </tr> </table>	31:0	<b>IER</b>		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.			<b>Value</b>	<b>Name</b>		0b	Disabled		1b	Enabled	
31:0	<b>IER</b>																		
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.																			
<b>Value</b>	<b>Name</b>																		
0b	Disabled																		
1b	Enabled																		

## Invalidate After Read Message Descriptor Control Field

<b>MDC_IAR - Invalidate After Read Message Descriptor Control Field</b>		
Size (in bits):		1
Default Value:		0x00000000
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0	0	<b>Reserved</b>
		Access: RO
		Format: MBZ

## JPEG

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



## LNCF\_POWER\_CONTEXT

LNCF_POWER_CONTEXT []				
Description	MMIO Offset/Command	Unit	# of DW	Address Offset(DWord)
LNCFMOCS0	0xB020		2	0x0000
LNCFMOCS1	0xB024		2	0x0002
LNCFMOCS2	0xB028		2	0x0004
LNCFMOCS3	0xB02C		2	0x0006
LNCFMOCS4	0xB030		2	0x0008
LNCFMOCS5	0xB034		2	0x000A
LNCFMOCS6	0xB038		2	0x000C
LNCFMOCS7	0xB03C		2	0x000E
LNCFMOCS8	0xB040		2	0x0010
LNCFMOCS9	0xB044		2	0x0012
LNCFMOCS10	0xB048		2	0x0014
LNCFMOCS11	0xB04C		2	0x0016
LNCFMOCS12	0xB050		2	0x0018
LNCFMOCS13	0xB054		2	0x001A
LNCFMOCS14	0xB058		2	0x001C
LNCFMOCS15	0xB05C		2	0x001E
LNCFMOCS16	0xB060		2	0x0020
LNCFMOCS17	0xB064		2	0x0022
LNCFMOCS18	0xB068		2	0x0024
LNCFMOCS19	0xB06C		2	0x0026
LNCFMOCS20	0xB070		2	0x0028
LNCFMOCS21	0xB074		2	0x002A
LNCFMOCS22	0xB078		2	0x002C
LNCFMOCS23	0xB07C		2	0x002E
LNCFMOCS24	0xB080		2	0x0030
LNCFMOCS25	0xB084		2	0x0032
LNCFMOCS26	0xB088		2	0x0034
LNCFMOCS27	0xB08C		2	0x0036
LNCFMOCS28	0xB090		2	0x0038
LNCFMOCS29	0xB094		2	0x003A
LNCFMOCS30	0xB098		2	0x003C
LNCFMOCS31	0xB09C		2	0x003E
SCRATCH_LNCF2	0xB0A0		2	0x0040

<b>LNCF_POWER_CONTEXT []</b>				
<b>Description</b>	<b>MMIO Offset/Command</b>	<b>Unit</b>	<b># of DW</b>	<b>Address Offset(DWord)</b>
SCRATCH_LNCF3	0xB0A8		2	0x0042
L3NODELOCK	0xB0AC		2	0x0044
LSN_SLCVC	0xB0C8		2	0x0046
LSN_MISC	0xB0CC		2	0x0048
LSN_ARBPRIO	0xB0D0		2	0x004A
LSN_ARBPRI1	0xB0D4		2	0x004C
GAFCFG0	0xD800		2	0x004E
GAFCFG1	0xD804		2	0x0050

## LOD Message Address Payload Control

MACD_LOD - LOD Message Address Payload Control								
Size (in bits):	32							
Default Value:	0x00000000							
DWord	Bit	Description						
0	31:4	<b>Reserved</b>						
		Access: RO						
		Format: MBZ						
	3:0	<b>LOD</b>						
		Format: U4						
		Specifies the LOD for this slot.						
		<table border="1"> <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]		representing LOD
Value	Name	Description						
[0,14]		representing LOD						



## Lower Oword Block Data Payload

<b>MDP_OW1L - Lower Oword Block Data Payload</b>						
Size (in bits):	256					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000					
DWord	Bit	Description				
0.0-0.3	127:0	<b>Oword</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U128</td> </tr> <tr> <td colspan="2">Specifies the upper Oword data element</td> </tr> </table>	Format:	U128	Specifies the upper Oword data element	
Format:	U128					
Specifies the upper Oword data element						
0.4-0.7	127:0	<b>Reserved</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
		Access:	RO			
Format:	MBZ					

## LPFC\_POWER\_CONTEXT

LPFC_POWER_CONTEXT []				
Description	MMIO Offset/Command	Unit	# of DW	Address Offset(DWord)
L3LPFCLOCK	0xB460		2	0x0000
SCRATCH_LPFC1	0xB474		2	0x0002



## LRI Data Entry

<b>LRI_DATA - LRI Data Entry</b>		
Source:	RenderCS	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
<p>Each LRI command header is followed by LRI_DATA entries. Each of these entries is a pair of Dwords: the MMIO register address and the data to be written.</p>		
DWord	Bit	Description
0..1	63:55	<b>Reserved</b>
		Access: RO
		Format: MBZ
	54:32	<b>MMIO</b>
		Format: U23
		<b>Programming Notes</b>
Bits [1:0] MBZ		
31:0	<b>Data</b>	
	Format: U32	

## MacroOperand

MacroOperand										
Size (in bits):		14								
Default Value:		0x00000000								
DWord	Bit	Description								
0	13:6	<b>RegNum</b> Format: U8 This field provide 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 destination or Source 0. Any Source 1 or Source 2 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. This field applies to both source and destination operands.								
	5	<b>Reserved</b> Access: RO Format: MBZ								
	4:1	<b>SpecialAccNum</b> This field specifies the accumulator numbers used by the IEEE macro instructions (madm and math.invm/math.rsqtm).The 8 special accumulators, acc2 to acc9 are encoded consecutively from 0000b to 0111b and noacc, indicating no special accumulator used is encoded as 1000b								
	0	<b>RegFile</b> This field indicate whether Architecture register file or General register file are selected. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>ARF</td> <td>Architecture Register File.Only allowed for Source 0 or Destination.</td> </tr> <tr> <td>1</td> <td>GRF</td> <td>General Register File. Allowed for any Source or Destination.</td> </tr> </tbody> </table>	Value	Name	Description	0	ARF	Architecture Register File.Only allowed for Source 0 or Destination.	1	GRF
Value	Name	Description								
0	ARF	Architecture Register File.Only allowed for Source 0 or Destination.								
1	GRF	General Register File. Allowed for any Source or Destination.								



## Manageability Engine Interrupt Vector

CSME_INTR_VEC - Manageability Engine Interrupt Vector		
Size (in bits):	16	
Default Value:	0x00000000	
DWord	Bit	Description
0	15:2	<b>Reserved</b>
		Access: RO
		Format: MBZ
	1	<b>CSME Response</b>
		Format: U1
0	<b>CSME Request</b>	
	Format: U1	
		CSME sets this bit in the interrupt when CSME initiates the transaction for: <ul style="list-style-type: none"> <li>• CSME to GuC wake up request</li> <li>• Payload message sent toto ME_MESG, ME_DATA for CMSE initiated request</li> </ul>



## MEDIA\_SURFACE\_STATE

MEDIA_SURFACE_STATE												
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.												
DWord	Bit	Description										
0	31:30	<b>Rotation</b>										
		<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>	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										
	<b>Programming Notes</b>											
	Rotation is only supported only with AVS function messages and not with HDC direct write and 16x8 AVS messages.											
	29:27	<b>Reserved</b>										
Access:		RO										
	Format:	MBZ										
26:20	<b>X Offset</b>											
	Exists If:	//[Surface Format] is one of Planar Formats										
	Format:	U7										
	This field specifies the horizontal offset in pixels from the <b>Surface Base Address</b> 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.											
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,127]</td> <td></td> <td>In multiples of 4 (low 2 bits missing)</td> </tr> </tbody> </table>	Value	Name	Description	[0,127]		In multiples of 4 (low 2 bits missing)					
	Value	Name	Description									
	[0,127]		In multiples of 4 (low 2 bits missing)									
	<b>Programming Notes</b>											
	For linear surfaces and Packed Formats, this field must be zero.											
	For <b>Surface Format</b> with 8 bits per element, this field must be a multiple of 16.											
For <b>Surface Format</b> with 16 bits per element, this field must be a multiple of 8.												

<b>MEDIA_SURFACE_STATE</b>												
	19:16	<p><b>Y Offset</b></p> <table border="1"> <tr> <td>Exists If:</td> <td>//[Surface Format] is one of Planar Formats</td> </tr> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>This field specifies the vertical offset in rows from the <b>Surface Base Address</b> to the start of the surface. (See additional description in the <b>X Offset</b> field)</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,15]</td> <td></td> <td>In multiples of 4 (low two bits missing)</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>For linear surfaces and Packed Formats, this field must be zero.</p>	Exists If:	//[Surface Format] is one of Planar Formats	Format:	U4	Value	Name	Description	[0,15]		In multiples of 4 (low two bits missing)
	Exists If:	//[Surface Format] is one of Planar Formats										
	Format:	U4										
	Value	Name	Description									
	[0,15]		In multiples of 4 (low two bits missing)									
	15:12	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
	Access:	RO										
	Format:	MBZ										
	11:0	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
	Access:	RO										
Format:	MBZ											
1	31:18	<p><b>Height</b></p> <table border="1"> <tr> <td>Format:</td> <td>U14-1</td> </tr> </table> <p>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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> <td>representing heights [1,16384]</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>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.</p>	Format:	U14-1	Value	Name	Description	[0,16383]		representing heights [1,16384]		
	Format:	U14-1										
	Value	Name	Description									
[0,16383]		representing heights [1,16384]										
17:4	<p><b>Width</b></p> <table border="1"> <tr> <td>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"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> <td>representing widths [1,16383]</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>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).</li> <li>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.</li> </ul>	Format:	U14-1	Value	Name	Description	[0,16383]		representing widths [1,16383]			
Format:	U14-1											
Value	Name	Description										
[0,16383]		representing widths [1,16383]										

## MEDIA\_SURFACE\_STATE

		<ul style="list-style-type: none"> <li>For deinterlace messages, the Width (field value + 1) must be a multiple of 8.</li> </ul>																																										
		Width (field value + 1) must be a multiple of 2 for PLANAR_420_16																																										
		For Y16_UNORM format width should be in multiple of 2																																										
	3:2	<p><b>Picture Structure</b> Specifies the encoding of the current picture.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Frame Picture</td> </tr> <tr> <td>01b</td> <td>Top Field Picture</td> </tr> <tr> <td>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																																											
	1:0	<p><b>Cr(V)/Cb(U) Pixel Offset V Direction</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U0.2</td> </tr> </table> <p>Specifies the distance to the U/V values with respect to the even numbered Y channels in the V direction</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This field is ignored for all formats except for PLANAR_420_8 and PLANAR_420_16</p> <p>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.</p>	Default Value:	0	Format:	U0.2																																						
Default Value:	0																																											
Format:	U0.2																																											
2	31:27	<p><b>Surface Format</b> Specifies the format of the surface. All of the Y and G channels will use table 0 and all of the Cr/Cb/R/B channels will use table 1.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>YCRCB_NORMAL</td> <td></td> </tr> <tr> <td>1</td> <td>YCRCB_SWAPUVY</td> <td></td> </tr> <tr> <td>2</td> <td>YCRCB_SWAPUV</td> <td></td> </tr> <tr> <td>3</td> <td>YCRCB_SWAPY</td> <td></td> </tr> <tr> <td>4</td> <td>PLANAR_420_8</td> <td></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> </tbody> </table>	Value	Name	Description	0	YCRCB_NORMAL		1	YCRCB_SWAPUVY		2	YCRCB_SWAPUV		3	YCRCB_SWAPY		4	PLANAR_420_8		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
Value	Name	Description																																										
0	YCRCB_NORMAL																																											
1	YCRCB_SWAPUVY																																											
2	YCRCB_SWAPUV																																											
3	YCRCB_SWAPY																																											
4	PLANAR_420_8																																											
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																																										

## MEDIA\_SURFACE\_STATE

	18	PLANAR_422_8	Sample_8x8 AVS only						
	23	PLANAR_420_16	Sample_8x8 AVS only						
	24	R16B16_UNORM (CrCb)	Sample_8x8 AVS only						
	25	R16_UNORM (Cr/Cb)	Sample_8x8 AVS only						
	26	Y16_UNORM	Sample_8x8 AVS only						
	Others	Reserved							
26	<b>Interleave Chroma</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field indicates that the chroma fields are interleaved in a single plane rather than stored as two separate planes. This field is only used for PLANAR surface formats.</p>			Format:	Enable				
Format:	Enable								
25	<b>Cr(V)/Cb(U) Pixel Offset U Direction</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U0.1</td> </tr> </table> <p>Specifies the distance to the U/V values with respect to the even numbered Y channels in the U direction</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>This field is must be zero for all formats except PLANAR_420_16, PLANAR_420_8, PLANAR_422_8, YCRCB_NORMAL, YCRCB_SWAPUVY, YCRCB_SWAPUV, YCRCB_SWAPY.</p>			Default Value:	0	Format:	U0.1	<b>Programming Notes</b>	
Default Value:	0								
Format:	U0.1								
<b>Programming Notes</b>									
24	<b>Cr(V)/Cb(U) Pixel Offset V Direction MSB</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Specifies the distance to the U/V values with respect to the even numbered Y channels in the V direction</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>This field is must be zero for all formats except?PLANAR_420_16 and 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>			Default Value:	0	Format:	U1	<b>Programming Notes</b>	
Default Value:	0								
Format:	U1								
<b>Programming Notes</b>									
23	<b>Memory Compression Type</b> Specifies the type of memory compression used. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr style="background-color: #e6f2ff;"> <th style="width: 20%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Media Compression</td> </tr> <tr> <td>1</td> <td>Render Compression</td> </tr> </tbody> </table>			Value	Name	0	Media Compression	1	Render Compression
Value	Name								
0	Media Compression								
1	Render Compression								
22	<b>Memory Compression Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">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>			Format:	Enable				
Format:	Enable								

## MEDIA\_SURFACE\_STATE

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	<b>Address Control</b>	
	<b>Value</b>	<b>Name</b>
	0	CLAMP
	1	MIRROR
20:3	<b>Surface Pitch</b>	
	Format:	U18-1
This field specifies the surface pitch in (#Bytes - 1).		
	<b>Value</b>	<b>Name</b>
	[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	<b>Half Pitch for Chroma</b>	
	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.		
1:0	<b>Tile Mode</b>	
	Format:	U2
This field specifies the type of memory tiling (Linear, WMajor, XMmajor, or YMajor) employed to tile this surface. See Memory Interface Functions for details on memory tiling and restrictions.		
	<b>Value</b>	<b>Name</b>
		<b>Description</b>

<b>MEDIA_SURFACE_STATE</b>														
		<table border="1"> <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> </table>	0h	TILEMODE_LINEAR	Linear mode (no tiling)	1h	Reserved	Reserved	2h	TILEMODE_XMAJOR	X major tiling	3h	TILEMODE_YMAJOR	Y major tiling
0h	TILEMODE_LINEAR	Linear mode (no tiling)												
1h	Reserved	Reserved												
2h	TILEMODE_XMAJOR	X major tiling												
3h	TILEMODE_YMAJOR	Y major tiling												
		<p style="text-align: center;"><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>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).</li> <li>The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this field.</li> <li>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.</li> </ul>												
3	31:30	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
	Access:	RO												
	Format:	MBZ												
	29:16	<p><b>X Offset for U(Cb)</b></p> <table border="1"> <tr> <td>Format:</td> <td>U14</td> </tr> </table> <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;"><b>Programming Notes</b></p> <p>For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.</p>	Format:	U14										
Format:	U14													
15:14	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ									
Access:	RO													
Format:	MBZ													
13:0	<p><b>Y Offset for U(Cb)</b></p> <table border="1"> <tr> <td>Format:</td> <td>U14</td> </tr> </table> <p>For non planar surfaces this field specifies the vertical offset in pixels from the Surface Base Address to the start (origin) of the surface.</p> <p>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.</p>	Format:	U14											
Format:	U14													

<b>MEDIA_SURFACE_STATE</b>																	
	<p>Resultant X-offset = 'Y-offset of the surface (Y-plane)' + 'Y 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;"><b>Programming Notes</b></p> <p>This field must be aligned by 4 bit[1:0] = 00</p> <p>This field must be aligned by 4 bit[1:0] = 00 for all format besides PLANAR_420_*</p>																
4	<p><b>31:30 Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p><b>29:16 X Offset for V(Cr)</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>///<math>[(\text{Surface Format}] \text{ is one of planar}) \text{ AND } ((\text{Interleave Chroma}] == '0')</math></td> </tr> <tr> <td>Format:</td> <td>U14</td> </tr> </table> <p>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)'</p> <p>For TileYS and TileYF this offset should be integral multiple of Tile width of Luma plane.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.</p> <p><b>15 Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p><b>14:0 Y Offset for V(Cr)</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>///<math>[(\text{Surface Format}] \text{ is one of planar}) \text{ AND } ((\text{Interleave Chroma}] == '0')</math></td> </tr> <tr> <td>Format:</td> <td>U15</td> </tr> </table> <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;"><b>Programming Notes</b></p> <p>This field must indicate a multiple of 4 (bit 0 &amp; 1 = 00).</p>	Access:	RO	Format:	MBZ	Exists If:	/// $[(\text{Surface Format}] \text{ is one of planar}) \text{ AND } ((\text{Interleave Chroma}] == '0')$	Format:	U14	Access:	RO	Format:	MBZ	Exists If:	/// $[(\text{Surface Format}] \text{ is one of planar}) \text{ AND } ((\text{Interleave Chroma}] == '0')$	Format:	U15
Access:	RO																
Format:	MBZ																
Exists If:	/// $[(\text{Surface Format}] \text{ is one of planar}) \text{ AND } ((\text{Interleave Chroma}] == '0')$																
Format:	U14																
Access:	RO																
Format:	MBZ																
Exists If:	/// $[(\text{Surface Format}] \text{ is one of planar}) \text{ AND } ((\text{Interleave Chroma}] == '0')$																
Format:	U15																
5	<p><b>31 Vertical Line Stride</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U1</td> </tr> </table> <p>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</p>	Format:	U1														
Format:	U1																

<b>MEDIA_SURFACE_STATE</b>																								
	Other Surfaces: Vertical Line Stride must be zero.																							
30	<p><b>Vertical Line Stride Offset</b></p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>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.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">This field must be set to 0 if Vertical Line Stride is 0.</td> </tr> </table>	Format:	U1	Programming Notes		This field must be set to 0 if Vertical Line Stride is 0.																		
Format:	U1																							
Programming Notes																								
This field must be set to 0 if Vertical Line Stride is 0.																								
29:20	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																			
Access:	RO																							
Format:	MBZ																							
19:18	<p><b>Tiled Resource Mode</b></p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p><b>For Sampling Engine, Render Target, and Typed/Untyped Surfaces:</b> This field specifies the tiled resource mode. <b>For other surfaces:</b> This field is ignored.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">If <b>Tile Mode</b> is not set to TILEMODE_YMAJOR, this field must be set to TRMODE_NONE.</td> </tr> <tr> <td colspan="2">If this field is not set to TRMODE_NONE, the <b>Surface Format</b> 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 <b>Interleave Chroma</b> is enabled) or 8 bits per element on the U and V planes (if <b>Interleave Chroma</b> is disabled).</td> </tr> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved		Programming Notes		If <b>Tile Mode</b> is not set to TILEMODE_YMAJOR, this field must be set to TRMODE_NONE.		If this field is not set to TRMODE_NONE, the <b>Surface Format</b> 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 <b>Interleave Chroma</b> is enabled) or 8 bits per element on the U and V planes (if <b>Interleave Chroma</b> is disabled).	
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																							
Programming Notes																								
If <b>Tile Mode</b> is not set to TILEMODE_YMAJOR, this field must be set to TRMODE_NONE.																								
If this field is not set to TRMODE_NONE, the <b>Surface Format</b> 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 <b>Interleave Chroma</b> is enabled) or 8 bits per element on the U and V planes (if <b>Interleave Chroma</b> is disabled).																								
17:7	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																			
Access:	RO																							
Format:	MBZ																							
6:0	<p><b>Surface Memory Object Control State</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0h DefaultVaueDesc</td> </tr> <tr> <td>Format:</td> <td><b>MEMORY_OBJECT_CONTROL_STATE</b></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:	<b>MEMORY_OBJECT_CONTROL_STATE</b>																			
Default Value:	0h DefaultVaueDesc																							
Format:	<b>MEMORY_OBJECT_CONTROL_STATE</b>																							



<b>MEDIA_SURFACE_STATE</b>						
6	31:0	<b>Surface Base Address</b> <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> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <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 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.</p>	Format:	GraphicsAddress[31:0]	<b>Programming Notes</b>	
		Format:	GraphicsAddress[31:0]			
<b>Programming Notes</b>						
<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
Access:	RO					
Format:	MBZ					
7	31:16	<b>Reserved</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					
15:0	15:0	<b>Surface Base Address High</b> <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						
Size (in bits):	7					
Default Value:	0x00000000					
DWord	Bit	Description				
0	6:1	<b>Index to MOCS Tables</b> 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. <table border="1"><thead><tr><th colspan="2">Programming Notes</th></tr></thead><tbody><tr><td colspan="2">When an access is made through Data Port and the index to MOCS[6:1] = [48,59] (decimal), that surface or stateless memory access can be cached in HDC L1 cache. Accesses made through Data Port with MOCS[6:1] &lt; 48 or &gt; 59 will bypass the HDC L1 cache. This bypass is useful when software wants to ensure that Data Port access are coherent with the L3 memory.</td></tr></tbody></table>	Programming Notes		When an access is made through Data Port and the index to MOCS[6:1] = [48,59] (decimal), that surface or stateless memory access can be cached in HDC L1 cache. Accesses made through Data Port with MOCS[6:1] < 48 or > 59 will bypass the HDC L1 cache. This bypass is useful when software wants to ensure that Data Port access are coherent with the L3 memory.	
Programming Notes						
When an access is made through Data Port and the index to MOCS[6:1] = [48,59] (decimal), that surface or stateless memory access can be cached in HDC L1 cache. Accesses made through Data Port with MOCS[6:1] < 48 or > 59 will bypass the HDC L1 cache. This bypass is useful when software wants to ensure that Data Port access are coherent with the L3 memory.						
	0	<b>Reserved</b>				

## MemoryAddressAttributes

MemoryAddressAttributes																	
Size (in bits):	32																
Default Value:	0x00000000																
This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface. It defines the attributes for VDBOX addresses.																	
DWord	Bit	Description															
0	31:15	<b>Reserved</b>															
		Access: RO															
		Format: MBZ															
14:13	14:13	<b>Base Address - Tiled Resource Mode</b>															
		Format: U2															
		<b>For Media Surfaces:</b> 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>00b</td> <td>TRMODE_NONE</td> <td>TileY resources</td> </tr> <tr> <td>01b</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>10b</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>11b</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	00b	TRMODE_NONE	TileY resources	01b	TRMODE_TILEYF	4KB tiled resources	10b	TRMODE_TILEYS	64KB tiled resources	11b	Reserved	
		Value	Name	Description													
		00b	TRMODE_NONE	TileY resources													
01b	TRMODE_TILEYF	4KB tiled resources															
10b	TRMODE_TILEYS	64KB tiled resources															
11b	Reserved																
12	12	<b>Base Address - Row Store Scratch Buffer Cache Select</b>															
		Format: U1															
		<p style="text-align: center;"><b>Description</b></p> <p>This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.</p> <p>When this is programmed to "1" (going to Media Cache), the corresponding base address will be programmed with the starting position in the media cache. The programming table is in "Buffer Size Requirement Page" in HEVC section</p>															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Buffer going to LLC.</td> </tr> <tr> <td>1</td> <td></td> <td>Buffer going to Internal Media Storage.</td> </tr> </tbody> </table>	Value	Name	Description	0		Buffer going to LLC.	1		Buffer going to Internal Media Storage.						
Value	Name	Description															
0		Buffer going to LLC.															
1		Buffer going to Internal Media Storage.															
11	11	<b>Reserved</b>															
		Access: RO															
		Format: MBZ															
10	10	<b>Compression Type</b> Indicates if buffer is render/media compressed.															

<b>MemoryAddressAttributes</b>		
	<b>Value</b>	<b>Name</b>
	0	Media Compression Enable <b>[Default]</b>
	1	Render Compression Enable
9	<b>Base Address - Memory Compression Enable</b>	
	Format:	Enable
	Memory compression will be attempted for this surface.	
8:7	<b>Base Address - Arbitration Priority Control</b>	
	Format:	<b>HEVC_ARBITRATION_PRIORITY</b>
6:1	<b>Base Address - Index to Memory Object Control State (MOCS) Tables</b>	
	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	<b>Reserved</b>	

## Merged Media Block Message Header Control

MHC_MBM_CONTROL - Merged Media Block Message Header Control											
Size (in bits):	32										
Default Value:	0x00000000										
DWord	Bit	Description									
0	31:30	<b>Message Mode</b> Specifies the Media Block Read message is Normal subtype.									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>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>Others</td> <td>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.									
29	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
Access:	RO										
Format:	MBZ										
28:24	<b>Sub-Register Offset</b> <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>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.</p> <table border="1"> <thead> <tr> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>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.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Restriction</th> </tr> </thead> <tbody> <tr> <td>For the Sampler Cache Data, this field must be zero.</td> </tr> <tr> <td>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.</td> </tr> <tr> <td>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.</td> </tr> </tbody> </table>	Format:	U5	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.		
Format:	U5										
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.											

## MHC\_MBM\_CONTROL - Merged Media Block Message Header Control

23:22	<b>Reserved</b>		
	Access:		RO
	Format:		MBZ
21:16	<b>Block Height</b>		
	Format:		U6
	Height in rows of block being accessed. Range = [0,63] representing 1 to 64 rows		
	<b>Restriction</b>		
	If Block Width (bytes), then Maximum Block Height (rows) is constrained by (# Dwords width) * (# rows) <= 64 Dwords.		
15:10	<b>Reserved</b>		
	Access:		RO
	Format:		MBZ
9:8	<b>Register Pitch Control</b>		
	Format:		U2
	Controls the register pitch for a Merged Media Block Read message. This field is ignored (reserved) for a media block write message. Register Pitch Control is only allowed to be non-zero when Block Width is a multiple of DWords.		
	Restriction : For the Sampler Cache Data, this field must be zero.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	RPC_1 <b>[Default]</b>	1 Block
	1h	RPC_2	2 Blocks
	3h	RPC_4	4 Blocks
	<b>Restriction</b>		
	BasePitch is defined as the next the power-of-2 that is greater than or equal to the Block Width. The effective register pitch (RPC*BasePitch)+SRO must be less than or equal to 32 bytes (to fit in a single GRF register).		
7:6	<b>Reserved</b>		
	Access:		RO
	Format:		MBZ
5:0	<b>Block Width</b>		
	Format:		U6
	Width in bytes of the block being accessed. Range = [0,31] representing 1 to 32 Bytes.		

## Message Descriptor - Render Target Write

Message Descriptor - Render Target Write				
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	30	<b>Data Format</b>		
		Format:	U1	
		Value	Name	Description
		0	Single Precision	32b
		1	Half Precision	16b
	<b>Programming Notes</b>			
This field is applicable for Render Target Write Messages ONLY.				
29:14	<b>Reserved</b>			
	Access:	RO		
	Format:	MBZ		
13	<b>Per-Sample PS outputs enable</b> 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	<b>Last Render Target Select</b> 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.			
	<b>Programming Notes</b>			
		In general, when threads are not launched by 3D FF, this bit must be zero.		
11	<b>Slot Group Select</b> 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			

## Message Descriptor - Render Target Write

		<p>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="width: 15%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 60%;">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																						
		<b>Programming Notes</b>																						
		For SIMD8 Image Write message thsi field MBZ.																						
10:8	<p><b>Message Type</b> 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" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>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_TM</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_TM	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_TM	It's only supported when accessing <i>Tiled Memory</i> . Using this Message Type to access linear ( <i>Untiled</i> ) memory is UNDEFINED.																						
		<b>Programming Notes</b>																						
		<p>the above slots indicated are within the 16 slots selected by <b>Slot Group Select</b>. 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.</p> <p>SIMD16_REPDATA message must not be used in SIMD8 pixel-shaders.</p>																						
7:0	<p><b>Reserved</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>			Access:	RO	Format:	MBZ																	
Access:	RO																							
Format:	MBZ																							



## Message Descriptor - Sampling Engine

Message Descriptor - Sampling Engine												
Size (in bits):	32											
Default Value:	0x00000000											
DWord	Bit	Description										
0	31	<b>EOT</b>										
	30	<b>Return Format</b>										
		Format:	U1									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>32-bit</td> <td>Return data is 32b</td> </tr> <tr> <td>1</td> <td>16-bit</td> <td>Return data is 16b</td> </tr> </tbody> </table>	Value	Name	Description	0	32-bit	Return data is 32b	1	16-bit	Return data is 16b	
		Value	Name	Description								
		0	32-bit	Return data is 32b								
	1	16-bit	Return data is 16b									
	<b>Programming Notes</b>											
	This field must be set to 32 for resinfo, LOD and sampleinfo messages.											
	29	<b>SIMD Mode[2]</b>										
Format:	U1											
This field is the upper bit of the 3-bit SIMD Mode field.												
28:25	<b>Message Length</b>											
	Format:	U4										
	This field specifies the number of 256-bit GRF registers starting from (src) to be sent out on the request message payload.											
24:20	<b>Response Length</b>	Format:	U5									
		This field indicates the number of 256-bit registers expected in the message response.										
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,16]							
		Value	Name									
		[0,16]										
	<b>Programming Notes</b>											
	A value 0 indicates that the request message does not expect any response. The largest response supported is 16 GRF registers.											
	This field must be programmed to 0 for a surface which is marked as a Procedural Texture (AMFS)											

## Message Descriptor - Sampling Engine

19	<b>Header Present</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Enable</td> </tr> </table> <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 <b>Write Channel Mask</b> fields are set according to the Response Length.</p> <p><b>SIMD16 or Return Format</b> Float16</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Response Length</th> <th>Mask R</th> <th>Mask G</th> <th>Mask B</th> <th>Mask A</th> </tr> </thead> <tbody> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>3</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table> <p><b>Else</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Response Length</th> <th>Mask R</th> <th>Mask G</th> <th>Mask B</th> <th>Mask A</th> </tr> </thead> <tbody> <tr><td>2</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>4</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>6</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	Format:	Enable	Response Length	Mask R	Mask G	Mask B	Mask A	1	0	1	1	1	2	0	0	1	1	3	0	0	0	1	4	0	0	0	0	Response Length	Mask R	Mask G	Mask B	Mask A	2	0	1	1	1	4	0	0	1	1	6	0	0	0	1	8	0	0	0	0
Format:	Enable																																																					
Response Length	Mask R	Mask G	Mask B	Mask A																																																		
1	0	1	1	1																																																		
2	0	0	1	1																																																		
3	0	0	0	1																																																		
4	0	0	0	0																																																		
Response Length	Mask R	Mask G	Mask B	Mask A																																																		
2	0	1	1	1																																																		
4	0	0	1	1																																																		
6	0	0	0	1																																																		
8	0	0	0	0																																																		
18:17	<b>SIMD Mode[1:0]</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U2</td> </tr> </table> <div style="background-color: #e6f2ff; padding: 5px; text-align: center; margin: 5px 0;"><b>Description</b></div> <p>Specifies the SIMD mode of the message being sent.</p> <p>A third bit SIMD Mode[2] is added to this field (bit 29 of message descriptor). Encodings now range from 0-7. SIMD Mode[2:0] SIMD</p> <ul style="list-style-type: none"> <li>000 Reserved</li> <li>001 SIMD8</li> <li>010 SIMD16</li> <li>011 SIMD32/64</li> <li>100 Reserved</li> <li>101 SIMD8H</li> <li>110 SIMD16H</li> <li>111 Reserved</li> </ul>	Format:	U2																																																		
Format:	U2																																																					
16:12	<b>Message Type</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U5</td> </tr> </table> <p>Specifies the type of message being sent. For more details, please refer to <b>Message Format</b> section for the definition of these 5 bits..</p>	Format:	U5																																																		
Format:	U5																																																					

<b>Message Descriptor - Sampling Engine</b>				
11:8	<b>Sampler Index</b>			
	Format: <span style="float: right;">U4</span>			
	Specifies the index into the sampler state table. Ignored for Id, resinfo, sampleinfo, and cache_flush type messages.			
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,15]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,15]
Value	Name			
[0,15]				
7:0	<b>Binding Table Index</b>			
	Format: <span style="float: right;">U8</span>			
	Specifies the index into the <b>binding table</b> . Ignored for cache_flush type messages. Values of 255 and 253 indicate stateless. 254 indicates SLM. 252 indicates bindless.			
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Value</th> <th style="width: 50%; text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,255]</td> <td></td> </tr> </tbody> </table>	Value	Name	[0,255]
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	<b>Slice Horizontal Position</b>									
		Format: <span style="float: right;">U8</span>									
		This field indicates the horizontal position of the first macroblock in the slice.									
	23:16	<b>Slice Vertical Position</b>									
Format: <span style="float: right;">U8</span> This field indicates the vertical position of the first macroblock in the slice.											
15:8	<b>Macroblock Count</b>										
	Format: <span style="float: right;">U8</span> This field indicates the number of macroblocks in the slice, including skipped macroblocks.										
7	7	<b>Slice Concealment Override Bit</b>									
		This bit forces hardware to handle the current slice in Conceal or Decode 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.									
		<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	6	<b>Slice Concealment Type Bit</b>									
		This bit can be forced by driver ("Slice Concealment Override Bit") or set by VINunit depending on slice boundary errors.									
		<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.									

## MFD\_MPEG2\_BSD\_OBJECT Inline Data Description

		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.		
5	<b>Last Pic Slice</b>			
	This bit is added to support error concealment at the end of a picture.			
	<b>Value</b>	<b>Name</b>	<b>Description</b>	
	1h		The current Slice is the last Slice of the entire picture	
	0h		The current Slice is not the last Slice of current picture	
	<b>Reserved</b>			
	3	<b>Is Last MB</b>		
		<b>Value</b>	<b>Name</b>	<b>Description</b>
		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	<b>First Macroblock Bit Offset</b>			
	Format:		U3	
This field provides the bit offset of the first macroblock in the first byte of the input bitstream.				
1	31:29	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	28:24	<b>Quantizer Scale Code</b>		
		Format:		U5
	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.			
	23:17	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	16:8	<b>Next Slice Vertical Position</b>		
Format:			U9	
This field indicates the vertical position (in macroblock units) of the first macroblock in the next slice.				
<b>Programming Notes</b>				
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).				

## MFD\_MPEG2\_BSD\_OBJECT Inline Data Description

	7:0	<p><b>Next Slice Horizontal Position</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U8</td> </tr> </table> <p>This field indicates the horizontal position (in macroblock units) of the first macroblock in the next slice.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>This field is primarily used for error concealment. In the case that current slice is the last slice, this field should set 0.</p>	Format:	U8	<b>Programming Notes</b>
Format:	U8				
<b>Programming Notes</b>					

## MFX\_REFERENCE\_PICTURE\_BASE\_ADDR

<b>MFX_REFERENCE_PICTURE_BASE_ADDR</b>		
Source:	VideoCS	
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:48	<b>Reserved</b>
		Access: RO
		Format: MBZ
	47:32	<b>MFX Reference Picture Address [n] High</b> Format: GraphicsAddress[47:32] This field is for the upper range of Reference Picture Addresses
31:6	31:6	<b>MFX Reference Picture Address [n]</b> Format: GraphicsAddress[31:6] Specifies the 64 byte aligned reference frame buffer addresses for the motion compensation operation in AVC//MPEG2. AVC can specify up to 16 YUV frame-based surfaces for both forward and backward references, i.e. L0+L1 total = 16 max. Any entry can be assigned to L0 or L1 or both lists. But VC1 and MPEG2, worst case, can use up to 2 YUV frame-based surfaces for both forward and backward references: <ul style="list-style-type: none"> <li>• P-MB : RefAddr[0] - temporal closest previous field of a reference frame (can be the current frame)</li> <li>• RefAddr[1]- next temporal closest previous field of a reference frame (must be different from the current frame)</li> </ul> It is a variant (without the LongTermRefPic specification) of the RefFrameList[16] defined in AVC DXVA Spec. RefAddr[0-15] is indexed by frame_storeID »1. It is not a packed list, i.e. invalid entries can scatter among the list. All invalid addresses must be set to a valid address RefAddr[0] by the driver. The same applies to VC1 and MPEG2.
		<b>Programming Notes</b>
		AVC: Always specifies all 16 addresses even some of them are not needed as indicated by the max num of active reference pictures. This is done for preventing data corruption (error, fault condition, etc.) by having all the references being set to a legal location.
	5:0	<b>Reserved</b> Access: RO Format: MBZ

## MPEG2

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



## MSAA Sample Number Message Address Control

<b>MACD_MSAA_SN - MSAA Sample Number Message Address Control</b>					
Size (in bits):		32			
Default Value:		0x00000000			
DWord	Bit	Description			
0	31:4	<b>Reserved</b>			
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
3:0	<b>Sample Number</b>				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U4</td> </tr> </table> <p>Specifies the sample number for the slot. If the sample number is larger than the <b>Number of Multisamples</b> in the Surface State, then the access is out of bounds.</p>	Format:	U4		
Format:	U4				

## MsgDesc

<b>MsgDesc</b>																
Source:	Eulsa															
Size (in bits):	32															
Default Value:	0x00000000															
DWord	Bit	Description														
0	31	<b>Reserved</b> <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														
	30	<b>Data Format</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U1</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th colspan="2" style="text-align: center; background-color: #e1eef6;">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">This field specifies the width of data read from sampler or written to render target.</td> </tr> <tr> <td colspan="2">This field specifies the width of data read from written to render target.</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 50%; text-align: center; background-color: #e1eef6;">Value</th> <th style="width: 50%; text-align: center; background-color: #e1eef6;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0b</td> <td style="text-align: center;">32bit</td> </tr> <tr> <td style="text-align: center;">1b</td> <td style="text-align: center;">16bit</td> </tr> </tbody> </table>	Format:	U1	Description		This field specifies the width of data read from sampler or written to render target.		This field specifies the width of data read from written to render target.		Value	Name	0b	32bit	1b	16bit
	Format:	U1														
	Description															
	This field specifies the width of data read from sampler or written to render target.															
	This field specifies the width of data read from written to render target.															
	Value	Name														
	0b	32bit														
	1b	16bit														
29	<b>SIMD Mode[2]</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">MBZ</td> </tr> </table> <p>This field is the upper bit of the 3-bit SIMD Mode Field. Refer to the SIMD Mode[1:0] Field for encodings</p>	Format:	MBZ													
Format:	MBZ															
28:25	<b>Message Length</b> This field specifies the number of GRF registers starting from <Src0.RegNum> to be sent out on the request message payload. Valid value ranges from 1 to 15. A value of 0 is considered erroneous. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 30%; text-align: center; background-color: #e1eef6;">Value</th> <th style="width: 70%; text-align: center; background-color: #e1eef6;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1-15</td> <td style="text-align: center;">Number of Registers</td> </tr> </tbody> </table>	Value	Name	1-15	Number of Registers											
Value	Name															
1-15	Number of Registers															
24:20	<b>Response Length</b> This field indicates the number of GRF 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. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 30%; text-align: center; background-color: #e1eef6;">Value</th> <th style="width: 70%; text-align: center; background-color: #e1eef6;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0-16</td> <td style="text-align: center;">Number of Registers</td> </tr> </tbody> </table>	Value	Name	0-16	Number of Registers											
Value	Name															
0-16	Number of Registers															

<b>MsgDesc</b>			
19	<p><b>Header Present</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">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><b>Function Control</b></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>		



## Named Barrier State

<b>NAMED_BAR_STATE - Named Barrier State</b>										
Size (in bits):	32									
Default Value:	0x00000000									
Hardware state of one "named barrier", which is part of the payload of Save_Barrier and Restore_Barrier messages.										
DWord	Bit	Description								
0	31:20	<b>Reserved</b> Format: <span style="float: right;">MBZ</span>								
	19	<b>Designated Thread</b> Designated thread for WMTP Save. <div style="border: 1px solid black; background-color: #e6f2ff; padding: 2px; text-align: center; margin: 5px 0;"><b>Programming Notes</b></div> This field is ignored for Restore_Barrier message payload. <div style="border: 1px solid black; background-color: #e6f2ff; padding: 2px; text-align: center; margin: 5px 0;"><b>Restriction</b></div> This field is valid only for Logical Barrier ID 0.								
	18:11	<b>Number of Consumers</b> Format: <span style="float: right;">U8</span> Specifies the number of consumer threads in the barrier.								
	10:3	<b>Number of Producers</b> Format: <span style="float: right;">U8</span> Specifies the number of producer threads in the barrier.								
	2:1	<b>Barrier Type</b> Format: <span style="float: right;">U2</span> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Producer_Consumer</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Producer_Only</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Consumer_Only</td> </tr> </tbody> </table>	Value	Name	0	Producer_Consumer	1	Producer_Only	2	Consumer_Only
	Value	Name								
0	Producer_Consumer									
1	Producer_Only									
2	Consumer_Only									
0	<b>Valid</b> Indicates this barrier's signal is set for the thread.									

## No Event Data Payload

<b>MDP_NO_EVENT - No Event Data Payload</b>						
Source:	EuSubFunctionGateway					
Size (in bits):	256					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000					
DWord	Bit	Description				
0..7	255:0	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					

## Normal Media Block Message Header

<b>MH_MB - Normal Media Block Message Header</b>		
Source:	EuSubFunctionDataPort1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	<b>X Offset</b>
		Format: <span style="float: right;">S31</span>
		X offset (in bytes) of the upper left corner of the block into the surface.
		<b>Programming Notes</b>
Must be DWord aligned (Bits 1:0 MBZ) for the write form of the message.		
1	31:0	<b>Y Offset</b>
		Format: <span style="float: right;">S31</span> Y offset (in rows) of the upper left corner of the block into the surface.
2	31:0	<b>Normal Media Block Message Control</b>
		Format: <span style="float: right;"><b>MHC_MB_CONTROL</b></span> Specifies the Normal message subtype and additional input parameters.
3	31:0	<b>Mask</b>
		Format: <span style="float: right;">U32</span> 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.
4	31:0	<b>FFTID</b>
		Format: <span style="float: right;"><b>MHC_FFTID</b></span> Fixed Function Thread ID
5..7	95:0	<b>Reserved</b>
		Access: <span style="float: right;">RO</span>
		Format: <span style="float: right;">MBZ</span>

## Normal Media Block Message Header Control

MHC_MB_CONTROL - Normal Media Block Message Header Control									
Size (in bits):		32							
Default Value:		0x00000000							
DWord	Bit	Description							
0	31:30	<b>Message Mode</b>							
		<b>Description</b>							
		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.							
		Specifies the interpretation of M0.3 (Pixel or Byte Mask).							
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>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>Others</td> <td>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
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.							
<b>Programming Notes</b>									
		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		<b>Reserved</b>							
		Access: RO							
		Format: MBZ							
28:24		<b>Sub-Register Offset</b>							
		Default Value: 0							
		Format: U5							
		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		<b>Reserved</b>							
		Access: RO							
		Format: MBZ							
21:16		<b>Block Height</b>							
		Format: U6							
		Height in rows of block being accessed. Range = [0,63] representing 1 to 64 rows							

## MHC\_MB\_CONTROL - Normal Media Block Message Header Control

		<b>Restriction</b>	
		If Block Width (bytes), then Maximum Block Height (rows) is constrained by (# Dwords width) * (# rows) <= 64 Dwords.	
15:10	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
9:8	<b>Register Pitch Control</b>		
	Default Value:	0	
	Format:	U2	
	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	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
5:0	<b>Block Width</b>		
	Format:	U6	
	<b>Description</b>		
	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.		
	<b>Programming Notes</b>		
	Must be DWord aligned for the write form of the message.		



## oMask Message Data Payload Register

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

## MDPR\_OMASK - oMask Message Data Payload Register

4	31:16	<b>oMask9</b> Format: <span style="float: right;">U16</span> oMask for Pixels [15:0] of Slot 9. Used only if Slot Group HI or SIMD16.
	15:0	<b>oMask8</b> Format: <span style="float: right;">U16</span> oMask for Pixels [15:0] of Slot 8. Used only if Slot Group HI or SIMD16.
5	31:16	<b>oMask11</b> Format: <span style="float: right;">U16</span> oMask for Pixels [15:0] of Slot 11. Used only if Slot Group HI or SIMD16.
	15:0	<b>oMask10</b> Format: <span style="float: right;">U16</span> oMask for Pixels [15:0] of Slot 10. Used only if Slot Group HI or SIMD16.
6	31:16	<b>oMask13</b> Format: <span style="float: right;">U16</span> oMask for Pixels [15:0] of Slot 13. Used only if Slot Group HI or SIMD16.
	15:0	<b>oMask12</b> Format: <span style="float: right;">U16</span> oMask for Pixels [15:0] of Slot 12. Used only if Slot Group HI or SIMD16.
7	31:16	<b>oMask15</b> Format: <span style="float: right;">U16</span> oMask for Pixels [15:0] of Slot 15. Used only if Slot Group HI or SIMD16.
	15:0	<b>oMask14</b> Format: <span style="float: right;">U16</span> oMask for Pixels [15:0] of Slot 14. Used only if Slot Group HI or SIMD16.

## OM Replicated SIMD16 Render Target Data Payload

MDP_RTW_M16REP - OM Replicated SIMD16 Render Target Data Payload		
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	<b>oMask</b>
		Format: <b>MDPR_OMASK</b> Slots [15:0] oMask
1.0-1.7	255:0	<b>RGBA</b>
		Format: <b>MDPR_RGBA</b> RGBA for all slots [15:0]



## OM S0A SIMD8 Render Target Data Payload

MDP_RTW_MA8 - OM S0A SIMD8 Render Target Data Payload		
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Source 0 Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	<b>oMask</b> Format: <b>MDPR_OMASK</b> Slots [7:0] oMask. Upper half ignored.
2.0-2.7	255:0	<b>Red</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Red
3.0-3.7	255:0	<b>Green</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Green
4.0-4.7	255:0	<b>Blue</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
5.0-5.7	255:0	<b>Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha





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



## MDP\_RTW\_M8DS - OM SIMD8 Dual Source Render Target Data Payload

6.0-6.7	255:0	<b>Src1 Green</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots[7:0] or [15:8] of Src1 Green
Format:	<b>MDP_DW_SIMD8</b>	
7.0-7.7	255:0	<b>Src1 Blue</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots[7:0] or [15:8] of Src1 Blue
Format:	<b>MDP_DW_SIMD8</b>	
8.0-8.7	255:0	<b>Src1 Alpha</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td style="text-align: center;"><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots[7:0] or [15:8] of Src1 Alpha
Format:	<b>MDP_DW_SIMD8</b>	



## OM SIMD8 Render Target Data Payload

MDP_RTW_M8 - OM SIMD8 Render Target Data Payload				
Size (in bits):	1280			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<b>oMask</b> <table border="1"> <tr> <td>Format:</td> <td><b>MDPR_OMASK</b></td> </tr> </table> Slots [7:0] oMask. Upper half ignored.	Format:	<b>MDPR_OMASK</b>
Format:	<b>MDPR_OMASK</b>			
1.0-1.7	255:0	<b>Red</b> <table border="1"> <tr> <td>Format:</td> <td><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots [7:0] Red	Format:	<b>MDP_DW_SIMD8</b>
Format:	<b>MDP_DW_SIMD8</b>			
2.0-2.7	255:0	<b>Green</b> <table border="1"> <tr> <td>Format:</td> <td><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots [7:0] Green	Format:	<b>MDP_DW_SIMD8</b>
Format:	<b>MDP_DW_SIMD8</b>			
3.0-3.7	255:0	<b>Blue</b> <table border="1"> <tr> <td>Format:</td> <td><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots [7:0] Blue	Format:	<b>MDP_DW_SIMD8</b>
Format:	<b>MDP_DW_SIMD8</b>			
4.0-4.7	255:0	<b>Alpha</b> <table border="1"> <tr> <td>Format:</td> <td><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots [7:0] Alpha	Format:	<b>MDP_DW_SIMD8</b>
Format:	<b>MDP_DW_SIMD8</b>			



MDP_RTW_M16 - OM SIMD16 Render Target Data Payload		
6.0-6.7	255:0	<b>Blue[15:8]</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots [15:8] Blue
Format:	<b>MDP_DW_SIMD8</b>	
7.0-7.7	255:0	<b>Alpha[7:0]</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots [7:0] Alpha
Format:	<b>MDP_DW_SIMD8</b>	
8.0-8.7	255:0	<b>Alpha[15:8]</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MDP_DW_SIMD8</b></td> </tr> </table> Slots [15:8] Alpha
Format:	<b>MDP_DW_SIMD8</b>	



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



## MDP\_RTW\_SM8DS - OS OM SIMD8 Dual Source Render Target Data Payload

5.0-5.7	255:0	<b>Src1 Red</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Red
6.0-6.7	255:0	<b>Src1 Green</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Green
7.0-7.7	255:0	<b>Src1 Blue</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Blue
8.0-8.7	255:0	<b>Src1 Alpha</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Alpha
9.0-9.7	255:0	<b>Stencil</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_STENCIL</span> Slots [7:0] or [15:8] of Stencil



## OS OM SIMD8 Render Target Data Payload

MDP_RTW_SM8 - OS OM SIMD8 Render Target Data Payload		
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>oMask</b> Format: <b>MDPR_OMASK</b> Slots [7:0] oMask. Upper half ignored.
1.0-1.7	255:0	<b>Red</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Red
2.0-2.7	255:0	<b>Green</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Green
3.0-3.7	255:0	<b>Blue</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
4.0-4.7	255:0	<b>Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha
5.0-5.7	255:0	<b>Stencil</b> Format: <b>MDPR_STENCIL</b> Slots [7:0] Stencil







## MDP\_RTW\_S8DS - OS SIMD8 Dual Source Render Target Data Payload

6.0-6.7	255:0	<b>Src1 Blue</b>	
		Format:	<b>MDP_DW_SIMD8</b>
Slots[7:0] or [15:8] of Src1 Blue			
7.0-7.7	255:0	<b>Src1 Alpha</b>	
		Format:	<b>MDP_DW_SIMD8</b>
Slots[7:0] or [15:8] of Src1 Alpha			
8.0-8.7	255:0	<b>Stencil</b>	
		Format:	<b>MDPR_STENCIL</b>
Slots [7:0] or [15:8] of Stencil			



## OS SIMD8 Render Target Data Payload

MDP_RTW_S8 - OS SIMD8 Render Target Data Payload		
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Red</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Red
1.0-1.7	255:0	<b>Green</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Green
2.0-2.7	255:0	<b>Blue</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
3.0-3.7	255:0	<b>Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha
4.0-4.7	255:0	<b>Stencil</b> Format: <b>MDPR_STENCIL</b> Slots [7:0] Stencil





## MDP\_RTW\_SZMA8 - OS SZ OM S0A SIMD8 Render Target Data Payload

6.0-6.7	255:0	<b>Source Depth</b>	
		Format:	<b>MDP_DW_SIMD8</b>
		Slots [7:0] Source Depth	
7.0-7.7	255:0	<b>Stencil</b>	
		Format:	<b>MDPR_STENCIL</b>
		Slots [7:0] Stencil	



## MDP\_RTW\_SZM8DS - OS SZ OM SIMD8 Dual Source Render Target Data Payload

5.0-5.7	255:0	<b>Src1 Red</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Red
6.0-6.7	255:0	<b>Src1 Green</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Green
7.0-7.7	255:0	<b>Src1 Blue</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Blue
8.0-8.7	255:0	<b>Src1 Alpha</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Alpha
9.0-9.7	255:0	<b>Source Depth</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] or [15:8] of Source Depth
10.0-10.7	255:0	<b>Stencil</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDPR_STENCIL</span> Slots [7:0] or [15:8] of Stencil



## OS SZ OM SIMD8 Render Target Data Payload

MDP_RTW_SZM8 - OS SZ OM SIMD8 Render Target Data Payload		
Size (in bits):	1792	
Default Value:	0x00000000, 0x00000000,	
DWord	Bit	Description
0.0-0.7	255:0	<b>oMask</b> Format: <b>MDPR_OMASK</b> Slots [7:0] oMask. Upper half ignored.
1.0-1.7	255:0	<b>Red</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Red
2.0-2.7	255:0	<b>Green</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Green
3.0-3.7	255:0	<b>Blue</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
4.0-4.7	255:0	<b>Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha
5.0-5.7	255:0	<b>Source Depth</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source Depth
6.0-6.7	255:0	<b>Stencil</b> Format: <b>MDPR_STENCIL</b> Slots [7:0] Stencil

## OS SZ S0A SIMD8 Render Target Data Payload

MDP_RTW_SZA8 - OS SZ S0A SIMD8 Render Target Data Payload		
Size (in bits):	1792	
Default Value:	0x00000000, 0x00000000,	
DWord	Bit	Description
0.0-0.7	255:0	<b>Source 0 Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	<b>Red</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Red
2.0-2.7	255:0	<b>Green</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Green
3.0-3.7	255:0	<b>Blue</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
4.0-4.7	255:0	<b>Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha
5.0-5.7	255:0	<b>Source Depth</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source Depth
6.0-6.7	255:0	<b>Stencil</b> Format: <b>MDPR_STENCIL</b> Slots [7:0] Stencil

## OS SZ SIMD8 Dual Source Render Target Data Payload

MDP_RTW_SZ8DS - OS SZ SIMD8 Dual Source Render Target Data Payload		
Size (in bits):	2560	
Default Value:	0x00000000, 0x00000000,	
DWord	Bit	Description
0.0-0.7	255:0	<b>Src0 Red</b> Format: <b>MDP_DW_SIMD8</b> Slots[7:0] or [15:8] of Src0 Red
1.0-1.7	255:0	<b>Src0 Green</b> Format: <b>MDP_DW_SIMD8</b> Slots[7:0] or [15:8] of Src0 Green
2.0-2.7	255:0	<b>Src0 Blue</b> Format: <b>MDP_DW_SIMD8</b> Slots[7:0] or [15:8] of Src0 Blue
3.0-3.7	255:0	<b>Src0 Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots[7:0] or [15:8] of Src0 Alpha
4.0-4.7	255:0	<b>Src1 Red</b> Format: <b>MDP_DW_SIMD8</b> Slots[7:0] or [15:8] of Src1 Red
5.0-5.7	255:0	<b>Src1 Green</b> Format: <b>MDP_DW_SIMD8</b> Slots[7:0] or [15:8] of Src1 Green

## MDP\_RTW\_SZ8DS - OS SZ SIMD8 Dual Source Render Target Data Payload

6.0-6.7	255:0	<b>Src1 Blue</b>	
		Format:	<b>MDP_DW_SIMD8</b>
Slots[7:0] or [15:8] of Src1 Blue			
7.0-7.7	255:0	<b>Src1 Alpha</b>	
		Format:	<b>MDP_DW_SIMD8</b>
Slots[7:0] or [15:8] of Src1 Alpha			
8.0-8.7	255:0	<b>Source Depth</b>	
		Format:	<b>MDP_DW_SIMD8</b>
Slots [7:0] or [15:8] of Source Depth			
9.0-9.7	255:0	<b>Stencil</b>	
		Format:	<b>MDPR_STENCIL</b>
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		
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Red</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Red
1.0-1.7	255:0	<b>Green</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Green
2.0-2.7	255:0	<b>Blue</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Blue
3.0-3.7	255:0	<b>Alpha</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Alpha
4.0-4.7	255:0	<b>Source Depth</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Source Depth
5.0-5.7	255:0	<b>Stencil</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_STENCIL</span> Slots [7:0] Stencil



## Oword 2 Block Data Payload

<b>MDP_OW2 - Oword 2 Block Data Payload</b>		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.3	127:0	<b>Oword0</b>
		<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	<b>Oword1</b>
		<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

<b>MDP_OW4 - Oword 4 Block Data Payload</b>		
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	<b>Data[1:0]</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_OW</b></td> </tr> </table> <p>Specifies the Oword data for block elements [1:0]</p>
Format:	<b>MDCR_OW</b>	
1.0-1.7	255:0	<b>Data[3:2]</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_OW</b></td> </tr> </table> <p>Specifies the Oword data for block elements [3:2]</p>
Format:	<b>MDCR_OW</b>	

## Oword 8 Block Data Payload

<b>MDP_OW8 - Oword 8 Block Data Payload</b>		
Size (in bits):	1024	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Data[1:0]</b>
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_OW</b></td> </tr> </table> <p>Specifies the Oword data for block elements [1:0]</p>
Format:	<b>MDCR_OW</b>	
1.0-1.7	255:0	<b>Data[3:2]</b>
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_OW</b></td> </tr> </table> <p>Specifies the Oword data for block elements [3:2]</p>
Format:	<b>MDCR_OW</b>	
2.0-2.7	255:0	<b>Data[5:4]</b>
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_OW</b></td> </tr> </table> <p>Specifies the Oword data for block elements [5:4]</p>
Format:	<b>MDCR_OW</b>	
3.0-3.7	255:0	<b>Data[7:6]</b>
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_OW</b></td> </tr> </table> <p>Specifies the Oword data for block elements [7:6]</p>
Format:	<b>MDCR_OW</b>	



## Oword A64 SIMD8 Atomic Operation CMPWR16B Message Data Payload

MDP_A64_AOP8_OW2 - Oword A64 SIMD8 Atomic Operation CMPWR16B Message Data Payload				
Size (in bits):	2048			
Default Value:	0x00000000, 0x00000000,			
DWord	Bit	Description		
0.0-0.7	255:0	<b>Slot[1:0] Src0</b> <table border="1" data-bbox="579 1026 1471 1079"> <tr> <td>Format:</td> <td>MDCR_OW</td> </tr> </table> Specifies the Slot [1:0] Source 0 data	Format:	MDCR_OW
Format:	MDCR_OW			
1.0-1.7	255:0	<b>Slot[3:2] Src0</b> <table border="1" data-bbox="579 1194 1471 1247"> <tr> <td>Format:</td> <td>MDCR_OW</td> </tr> </table> Specifies the Slot [3:2] Source 0 data	Format:	MDCR_OW
Format:	MDCR_OW			
2.0-2.7	255:0	<b>Slot[5:4] Src0</b> <table border="1" data-bbox="579 1362 1471 1415"> <tr> <td>Format:</td> <td>MDCR_OW</td> </tr> </table> Specifies the Slot [5:4] Source 0 data	Format:	MDCR_OW
Format:	MDCR_OW			
3.0-3.7	255:0	<b>Slot[7:6] Src0</b> <table border="1" data-bbox="579 1530 1471 1583"> <tr> <td>Format:</td> <td>MDCR_OW</td> </tr> </table> Specifies the Slot [7:6] Source 0 data	Format:	MDCR_OW
Format:	MDCR_OW			
4.0-4.7	255:0	<b>Slot[1:0] Src1</b> <table border="1" data-bbox="579 1698 1471 1751"> <tr> <td>Format:</td> <td>MDCR_OW</td> </tr> </table> Specifies the Slot [1:0] Source 1 data	Format:	MDCR_OW
Format:	MDCR_OW			
5.0-5.7	255:0	<b>Slot[3:2] Src1</b> <table border="1" data-bbox="579 1866 1471 1919"> <tr> <td>Format:</td> <td>MDCR_OW</td> </tr> </table> Specifies the Slot [3:2] Source 1 data	Format:	MDCR_OW
Format:	MDCR_OW			

<b>MDP_A64_AOP8_OW2 - Oword A64 SIMD8 Atomic Operation CMPWR16B Message Data Payload</b>				
6.0-6.7	255:0	<b>Slot[5:4] Src1</b> <table border="1"> <tr> <td>Format:</td> <td><b>MDCR_OW</b></td> </tr> </table> Specifies the Slot [5:4] Source 1 data	Format:	<b>MDCR_OW</b>
Format:	<b>MDCR_OW</b>			
7.0-7.7	255:0	<b>Slot[7:6] Src1</b> <table border="1"> <tr> <td>Format:</td> <td><b>MDCR_OW</b></td> </tr> </table> Specifies the Slot [7:6] Source 1 data	Format:	<b>MDCR_OW</b>
Format:	<b>MDCR_OW</b>			

## Oword Data Blocks Message Descriptor Control Field

<b>MDC_DB_OW - Oword Data Blocks Message Descriptor Control Field</b>																							
Size (in bits): 3																							
Default Value: 0x00000000																							
DWord	Bit	Description																					
0	2:0	<b>Data Blocks</b> Specifies the number of Oword blocks to be read or written																					
		<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;">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;">Reserved</td> <td>Reserved</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> </tbody> </table>	Value	Name	Description	00h	OW1L	1 Oword, read into or written from the low 128 bits of the destination register	01h	Reserved	Reserved	02h	OW2	2 Owords	03h	OW4	4 Owords	04h	OW8	8 Owords	Others	Reserved	Ignored
Value	Name	Description																					
00h	OW1L	1 Oword, read into or written from the low 128 bits of the destination register																					
01h	Reserved	Reserved																					
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		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.3	127:0	<b>Oword0</b>
		Format: U128 Specifies the slot 0 data in this payload register
0.4-0.7	127:0	<b>Oword1</b>
		Format: U128 Specifies the slot 1 data in this payload register

## Oword Dual Data Blocks Message Descriptor Control Field

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



## Performance Counter Report Format 101b

Performance Counter Report Format 101b		
Size (in bits):	2048	
Default Value:	0x00000000, 0x00000000,	
DWord	Bit	Description
0	31:0	RPT_ID
1	31:0	TIME_STAMP
2	31:0	CTX_ID
3	31:0	GPU_TICKS
4	31:0	A-Cntr 0 (low dword)
5	31:0	A-Cntr 1 (low dword)
6	31:0	A-Cntr 2 (low dword)
7	31:0	A-Cntr 3 (low dword)
8	31:0	A-Cntr 4 (low dword)
9	31:0	A-Cntr 5 (low dword)
10	31:0	A-Cntr 6 (low dword)
11	31:0	A-Cntr 7 (low dword)
12	31:0	A-Cntr 8 (low dword)
13	31:0	A-Cntr 9 (low dword)
14	31:0	A-Cntr 10 (low dword)
15	31:0	A-Cntr 11 (low dword)
16	31:0	A-Cntr 12 (low dword)
17	31:0	A-Cntr 13 (low dword)
18	31:0	A-Cntr 14 (low dword)
19	31:0	A-Cntr 15 (low dword)
20	31:0	A-Cntr 16 (low dword)
21	31:0	A-Cntr 17 (low dword)
22	31:0	A-Cntr 18 (low dword)

## Performance Counter Report Format 101b

23	31:0	<b>A-Cntr 19 (low dword)</b>
24	31:0	<b>A-Cntr 20 (low dword)</b>
25	31:0	<b>A-Cntr 21 (low dword)</b>
26	31:0	<b>A-Cntr 22 (low dword)</b>
27	31:0	<b>A-Cntr 23 (low dword)</b>
28	31:0	<b>A-Cntr 24 (low dword)</b>
29	31:0	<b>A-Cntr 25 (low dword)</b>
30	31:0	<b>A-Cntr 26 (low dword)</b>
31	31:0	<b>A-Cntr 27 (low dword)</b>
32	31:0	<b>A-Cntr 28 (low dword)</b>
33	31:0	<b>A-Cntr 29 (low dword)</b>
34	31:0	<b>A-Cntr 30 (low dword)</b>
35	31:0	<b>A-Cntr 31 (low dword)</b>
36	31:0	<b>A-Cntr 32 (low dword)</b>
37	31:0	<b>A-Cntr 33 (low dword)</b>
38	31:0	<b>A-Cntr 34 (low dword)</b>
39	31:0	<b>A-Cntr 35 (low dword)</b>
40	31:24	<b>High byte of A3</b>
	23:16	<b>High byte of A2</b>
	15:8	<b>High byte of A1</b>
	7:0	<b>High byte of A0</b>
41	31:24	<b>High byte of A7</b>
	23:16	<b>High byte of A6</b>
	15:8	<b>High byte of A5</b>
	7:0	<b>High byte of A4</b>
42	31:24	<b>High byte of A11</b>
	23:16	<b>High byte of A10</b>
	15:8	<b>High byte of A9</b>
	7:0	<b>High byte of A8</b>
43	31:24	<b>High byte of A15</b>
	23:16	<b>High byte of A14</b>
	15:8	<b>High byte of A13</b>
	7:0	<b>High byte of A12</b>
44	31:24	<b>High byte of A19</b>
	23:16	<b>High byte of A18</b>
	15:8	<b>High byte of A17</b>

## Performance Counter Report Format 101b

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



## Per Thread Scratch Space Message Header Control

<b>MHC_PTSS - Per Thread Scratch Space Message Header Control</b>				
Size (in bits):		32		
Default Value:		0x00000000		
<b>DWord</b>	<b>Bit</b>	<b>Description</b>		
0	31:4	<b>Reserved</b>		
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
	3:0	<b>Per Thread Scratch Space</b>		
		<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>	Format:	U4
		Format:	U4	
<table border="1"> <tr> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="2">Writes out of bounds will be ignored. Reads out of bounds will return 0.</td> </tr> </table>	<b>Programming Notes</b>		Writes out of bounds will be ignored. Reads out of bounds will return 0.	
<b>Programming Notes</b>				
Writes out of bounds will be ignored. Reads out of bounds will return 0.				

## PIXEL\_HASH\_TABLE\_1BIT\_32ENTRY

PIXEL_HASH_TABLE_1BIT_32ENTRY				
Size (in bits):	32			
Default Value:	0x00000000			
Description				
2-way pixel hashing table. Table is 32-entries:8X,4Y in [Y][X] format. Each entry is a single bit that indicates which sub-slice hardware block the indicated xy pixel block is mapped.				
pixelhash_id maps to color-pipe. A value of 0 indicates the larger color-pipe, or first enabled color-pipe if both enabled color-pipes are balanced				
DWord	Bit	Description		
0	31:24	<b>Pixel Hashing Table Entries y[3]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has y=3 and x=7..0	Format:	U8
	Format:	U8		
	23:16	<b>Pixel Hashing Table Entries y[2]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has y=2 and x=7..0	Format:	U8
	Format:	U8		
15:8	<b>Pixel Hashing Table Entries y[1]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has y=1 and x=7..0	Format:	U8	
Format:	U8			
7:0	<b>Pixel Hashing Table Entries y[0]x[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has y=0 and x=7..0	Format:	U8	
Format:	U8			

## PIXEL\_HASH\_TABLE\_1BIT\_64ENTRY

PIXEL_HASH_TABLE_1BIT_64ENTRY				
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
Description				
2-way pixel hashing table. Table is 64-entries:8X,8Y in [Y][X] format. Each entry is a single bit that indicates which sub-slice hardware block the indicated xy pixel block is mapped.				
pixelhash_id maps to color-pipe. A value of 0 indicates the larger color-pipe, or first enabled color-pipe if both enabled color-pipes are balanced				
DWord	Bit	Description		
0	31:24	<b>Pixel Hashing Table Entries y[3]x[7:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=3 and x=7..0		U8
		U8		
	23:16	<b>Pixel Hashing Table Entries y[2]x[7:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=2 and x=7..0		U8
		U8		
15:8	<b>Pixel Hashing Table Entries y[1]x[7:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=1 and x=7..0		U8	
	U8			
7:0	<b>Pixel Hashing Table Entries y[0]x[7:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=0 and x=7..0		U8	
	U8			
1	31:24	<b>Pixel Hashing Table Entries y[7]x[7:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=7 and x=7..0		U8
		U8		
	23:16	<b>Pixel Hashing Table Entries y[6]x[7:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=6 and x=7..0		U8
		U8		
15:8	<b>Pixel Hashing Table Entries y[5]x[7:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=5 and x=7..0		U8	
	U8			
7:0	<b>Pixel Hashing Table Entries y[4]x[7:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U8</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=4 and x=7..0		U8	
	U8			



## PIXEL\_HASH\_TABLE\_1BIT\_128ENTRY

PIXEL_HASH_TABLE_1BIT_128ENTRY				
Size (in bits):	128			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000			
Description				
2-way pixel hashing table. Table is 128-entries:16X,8Y in [Y][X] format. Each entry is a single bit that indicates which sub-slice hardware block the indicated xy pixel block is mapped.				
pixelhash_id maps to color-pipe. A value of 0 indicates the larger color-pipe, or first enabled color-pipe if both enabled color-pipes are balanced				
DWord	Bit	Description		
0	31:16	<b>Pixel Hashing Table Entries y[1]x[15:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U16</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=1 and x=15..0		U16
		U16		
15:0	<b>Pixel Hashing Table Entries y[0]x[15:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U16</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=0 and x=15..0		U16	
	U16			
1	31:16	<b>Pixel Hashing Table Entries y[3]x[15:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U16</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=3 and x=15..0		U16
		U16		
15:0	<b>Pixel Hashing Table Entries y[2]x[15:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U16</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=2 and x=15..0		U16	
	U16			
2	31:16	<b>Pixel Hashing Table Entries y[5]x[15:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U16</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=5 and x=15..0		U16
		U16		
15:0	<b>Pixel Hashing Table Entries y[4]x[15:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U16</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=4 and x=15..0		U16	
	U16			
3	31:16	<b>Pixel Hashing Table Entries y[7]x[15:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U16</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=7 and x=15..0		U16
		U16		
15:0	<b>Pixel Hashing Table Entries y[6]x[15:0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U16</td></tr></table> Indicates the pixelhash_id for the pixel block that has y=6 and x=15..0		U16	
	U16			

## PIXEL\_HASH\_TABLE\_2BIT\_64ENTRY

PIXEL_HASH_TABLE_2BIT_64ENTRY				
Size (in bits):	128			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000			
Description				
<p>3-wayor 4-way pixel hashing table. Table is 64-entries:8X,8Y in [Y][X] format. Each entry is two bits that indicates which sub-slice hardware block the indicated xy pixel block is mapped.</p> <p>pixelhash_id maps to color-pipe. A value of 0 indicates the largest color-pipe, or first enabled color-pipe if all enabled color-pipes are balanced. A value of 2 indicates the smallest color-pipe, or last enabled color-pipe if all enabled color-pipes are balanced.</p>				
DWord	Bit	Description		
0	31:30	<b>Pixel Hashing Table Entry y[1]x[7]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=7 and y=1		U2
		U2		
	29:28	<b>Pixel Hashing Table Entry y[1]x[6]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=6 and y=1		U2
		U2		
	27:26	<b>Pixel Hashing Table Entry y[1]x[5]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=5 and y=1		U2
		U2		
	25:24	<b>Pixel Hashing Table Entry y[1]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=1		U2
		U2		
23:22	<b>Pixel Hashing Table Entry y[1]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=1		U2	
	U2			
21:20	<b>Pixel Hashing Table Entry y[1]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=1		U2	
	U2			
19:18	<b>Pixel Hashing Table Entry y[1]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=1		U2	
	U2			

<b>PIXEL_HASH_TABLE_2BIT_64ENTRY</b>				
	17:16	<b>Pixel Hashing Table Entry y[1]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=1	Format:	U2
	Format:	U2		
	15:14	<b>Pixel Hashing Table Entry y[0]x[7]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=7 and y=0	Format:	U2
	Format:	U2		
	13:12	<b>Pixel Hashing Table Entry y[0]x[6]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=6 and y=0	Format:	U2
	Format:	U2		
	11:10	<b>Pixel Hashing Table Entry y[0]x[5]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=5 and y=0	Format:	U2
	Format:	U2		
	9:8	<b>Pixel Hashing Table Entry y[0]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=0	Format:	U2
Format:	U2			
7:6	<b>Pixel Hashing Table Entry y[0]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=0	Format:	U2	
Format:	U2			
5:4	<b>Pixel Hashing Table Entry y[0]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=0	Format:	U2	
Format:	U2			
3:2	<b>Pixel Hashing Table Entry y[0]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=0	Format:	U2	
Format:	U2			
1:0	<b>Pixel Hashing Table Entry y[0]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=0	Format:	U2	
Format:	U2			
1	31:30	<b>Pixel Hashing Table Entry y[3]x[7]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=7 and y=3	Format:	U2
Format:	U2			

<b>PIXEL_HASH_TABLE_2BIT_64ENTRY</b>			
29:28	<b>Pixel Hashing Table Entry y[3]x[6]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=6 and y=3</p>	Format:	U2
Format:	U2		
27:26	<b>Pixel Hashing Table Entry y[3]x[5]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=5 and y=3</p>	Format:	U2
Format:	U2		
25:24	<b>Pixel Hashing Table Entry y[3]x[4]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=4 and y=3</p>	Format:	U2
Format:	U2		
23:22	<b>Pixel Hashing Table Entry y[3]x[3]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=3 and y=3</p>	Format:	U2
Format:	U2		
21:20	<b>Pixel Hashing Table Entry y[3]x[2]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=2 and y=3</p>	Format:	U2
Format:	U2		
19:18	<b>Pixel Hashing Table Entry y[3]x[1]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=1 and y=3</p>	Format:	U2
Format:	U2		
17:16	<b>Pixel Hashing Table Entry y[3]x[0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=0 and y=3</p>	Format:	U2
Format:	U2		
15:14	<b>Pixel Hashing Table Entry y[2]x[7]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=7 and y=2</p>	Format:	U2
Format:	U2		
13:12	<b>Pixel Hashing Table Entry y[2]x[6]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=6 and y=2</p>	Format:	U2
Format:	U2		
11:10	<b>Pixel Hashing Table Entry y[2]x[5]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=5 and y=2</p>	Format:	U2
Format:	U2		

PIXEL_HASH_TABLE_2BIT_64ENTRY				
	9:8	<b>Pixel Hashing Table Entry y[2]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=2		U2
		U2		
	7:6	<b>Pixel Hashing Table Entry y[2]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=2		U2
		U2		
	5:4	<b>Pixel Hashing Table Entry y[2]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=2		U2
	U2			
3:2	<b>Pixel Hashing Table Entry y[2]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=2		U2	
	U2			
1:0	<b>Pixel Hashing Table Entry y[2]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=2		U2	
	U2			
2	31:30	<b>Pixel Hashing Table Entry y[5]x[7]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=7 and y=5		U2
		U2		
	29:28	<b>Pixel Hashing Table Entry y[5]x[6]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=6 and y=5		U2
		U2		
	27:26	<b>Pixel Hashing Table Entry y[5]x[5]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=5 and y=5		U2
	U2			
25:24	<b>Pixel Hashing Table Entry y[5]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=5		U2	
	U2			
23:22	<b>Pixel Hashing Table Entry y[5]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=5		U2	
	U2			



<b>PIXEL_HASH_TABLE_2BIT_64ENTRY</b>			
21:20	<p><b>Pixel Hashing Table Entry y[5]x[2]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=2 and y=5</p>	Format:	U2
Format:	U2		
19:18	<p><b>Pixel Hashing Table Entry y[5]x[1]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=1 and y=5</p>	Format:	U2
Format:	U2		
17:16	<p><b>Pixel Hashing Table Entry y[5]x[0]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=0 and y=5</p>	Format:	U2
Format:	U2		
15:14	<p><b>Pixel Hashing Table Entry y[4]x[7]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=7 and y=4</p>	Format:	U2
Format:	U2		
13:12	<p><b>Pixel Hashing Table Entry y[4]x[6]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=6 and y=4</p>	Format:	U2
Format:	U2		
11:10	<p><b>Pixel Hashing Table Entry y[4]x[5]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=5 and y=4</p>	Format:	U2
Format:	U2		
9:8	<p><b>Pixel Hashing Table Entry y[4]x[4]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=4 and y=4</p>	Format:	U2
Format:	U2		
7:6	<p><b>Pixel Hashing Table Entry y[4]x[3]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=3 and y=4</p>	Format:	U2
Format:	U2		
5:4	<p><b>Pixel Hashing Table Entry y[4]x[2]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=2 and y=4</p>	Format:	U2
Format:	U2		
3:2	<p><b>Pixel Hashing Table Entry y[4]x[1]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=1 and y=4</p>	Format:	U2
Format:	U2		

<b>PIXEL_HASH_TABLE_2BIT_64ENTRY</b>				
	1:0	<b>Pixel Hashing Table Entry y[4]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=4	Format:	U2
Format:	U2			
3	31:30	<b>Pixel Hashing Table Entry y[7]x[7]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=7 and y=7	Format:	U2
	Format:	U2		
	29:28	<b>Pixel Hashing Table Entry y[7]x[6]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=6 and y=7	Format:	U2
	Format:	U2		
	27:26	<b>Pixel Hashing Table Entry y[7]x[5]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=5 and y=7	Format:	U2
	Format:	U2		
	25:24	<b>Pixel Hashing Table Entry y[7]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=7	Format:	U2
	Format:	U2		
	23:22	<b>Pixel Hashing Table Entry y[7]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=7	Format:	U2
Format:	U2			
21:20	<b>Pixel Hashing Table Entry y[7]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=7	Format:	U2	
Format:	U2			
19:18	<b>Pixel Hashing Table Entry y[7]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=7	Format:	U2	
Format:	U2			
17:16	<b>Pixel Hashing Table Entry y[7]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=7	Format:	U2	
Format:	U2			
15:14	<b>Pixel Hashing Table Entry y[6]x[7]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=7 and y=6	Format:	U2	
Format:	U2			

<b>PIXEL_HASH_TABLE_2BIT_64ENTRY</b>			
13:12	<p><b>Pixel Hashing Table Entry y[6]x[6]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=6 and y=6</p>	Format:	U2
Format:	U2		
11:10	<p><b>Pixel Hashing Table Entry y[6]x[5]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=5 and y=6</p>	Format:	U2
Format:	U2		
9:8	<p><b>Pixel Hashing Table Entry y[6]x[4]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=4 and y=6</p>	Format:	U2
Format:	U2		
7:6	<p><b>Pixel Hashing Table Entry y[6]x[3]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=3 and y=6</p>	Format:	U2
Format:	U2		
5:4	<p><b>Pixel Hashing Table Entry y[6]x[2]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=2 and y=6</p>	Format:	U2
Format:	U2		
3:2	<p><b>Pixel Hashing Table Entry y[6]x[1]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=1 and y=6</p>	Format:	U2
Format:	U2		
1:0	<p><b>Pixel Hashing Table Entry y[6]x[0]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=0 and y=6</p>	Format:	U2
Format:	U2		



## PIXEL\_HASH\_TABLE\_2BIT\_128ENTRY

PIXEL_HASH_TABLE_2BIT_128ENTRY				
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
Description				
3-way or 4-way pixel hashing table. Table is 128-entries:16X,8Y in [Y][X] format. Each entry is two bits that indicates which sub-slice hardware block the indicated xy pixel block is mapped.				
pixelhash_id maps to color-pipe. A value of 0 indicates the largest color-pipe, or first enabled color-pipe if all enabled color-pipes are balanced. A value of 2 indicates the smallest color-pipe, or last enabled color-pipe if all enabled color-pipes are balanced.				
DWord	Bit	Description		
0	31:30	<b>Pixel Hashing Table Entry y[0]x[15]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=15 and y=0		U2
		U2		
	29:28	<b>Pixel Hashing Table Entry y[0]x[14]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=14 and y=0		U2
		U2		
	27:26	<b>Pixel Hashing Table Entry y[0]x[13]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=13 and y=0		U2
		U2		
	25:24	<b>Pixel Hashing Table Entry y[0]x[12]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=12 and y=0		U2
		U2		
23:22	<b>Pixel Hashing Table Entry y[0]x[11]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=11 and y=0		U2	
	U2			
21:20	<b>Pixel Hashing Table Entry y[0]x[10]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=10 and y=0		U2	
	U2			
19:18	<b>Pixel Hashing Table Entry y[0]x[9]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=9 and y=0		U2	
	U2			

<b>PIXEL_HASH_TABLE_2BIT_128ENTRY</b>				
	17:16	<b>Pixel Hashing Table Entry y[0]x[8]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=8 and y=0		U2
		U2		
	15:14	<b>Pixel Hashing Table Entry y[0]x[7]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=7 and y=0		U2
		U2		
	13:12	<b>Pixel Hashing Table Entry y[0]x[6]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=6 and y=0		U2
		U2		
	11:10	<b>Pixel Hashing Table Entry y[0]x[5]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=5 and y=0		U2
		U2		
	9:8	<b>Pixel Hashing Table Entry y[0]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=0		U2
	U2			
7:6	<b>Pixel Hashing Table Entry y[0]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=0		U2	
	U2			
5:4	<b>Pixel Hashing Table Entry y[0]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=0		U2	
	U2			
3:2	<b>Pixel Hashing Table Entry y[0]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=0		U2	
	U2			
1:0	<b>Pixel Hashing Table Entry y[0]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=0		U2	
	U2			
1	31:30	<b>Pixel Hashing Table Entry y[1]x[15]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=15 and y=1		U2
	U2			

## PIXEL\_HASH\_TABLE\_2BIT\_128ENTRY

	29:28	<b>Pixel Hashing Table Entry y[1]x[14]</b>
		Format: U2
		Indicates the pixelhash_id for the pixel block that has x=14 and y=1
	27:26	<b>Pixel Hashing Table Entry y[1]x[13]</b>
		Format: U2
		Indicates the pixelhash_id for the pixel block that has x=13 and y=1
	25:24	<b>Pixel Hashing Table Entry y[1]x[12]</b>
		Format: U2
		Indicates the pixelhash_id for the pixel block that has x=12 and y=1
	23:22	<b>Pixel Hashing Table Entry y[1]x[11]</b>
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=11 and y=1	
21:20	<b>Pixel Hashing Table Entry y[1]x[10]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=10 and y=1	
19:18	<b>Pixel Hashing Table Entry y[1]x[9]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=9 and y=1	
17:16	<b>Pixel Hashing Table Entry y[1]x[8]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=8 and y=1	
15:14	<b>Pixel Hashing Table Entry y[1]x[7]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=7 and y=1	
13:12	<b>Pixel Hashing Table Entry y[1]x[6]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=6 and y=1	
11:10	<b>Pixel Hashing Table Entry y[1]x[5]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=5 and y=1	

<b>PIXEL_HASH_TABLE_2BIT_128ENTRY</b>				
	9:8	<b>Pixel Hashing Table Entry y[1]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=1	Format:	U2
	Format:	U2		
	7:6	<b>Pixel Hashing Table Entry y[1]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=1	Format:	U2
	Format:	U2		
	5:4	<b>Pixel Hashing Table Entry y[1]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=1	Format:	U2
Format:	U2			
3:2	<b>Pixel Hashing Table Entry y[1]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=1	Format:	U2	
Format:	U2			
1:0	<b>Pixel Hashing Table Entry y[1]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=1	Format:	U2	
Format:	U2			
2	31:30	<b>Pixel Hashing Table Entry y[2]x[15]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=15 and y=2	Format:	U2
	Format:	U2		
	29:28	<b>Pixel Hashing Table Entry y[2]x[14]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=14 and y=2	Format:	U2
	Format:	U2		
	27:26	<b>Pixel Hashing Table Entry y[2]x[13]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=13 and y=2	Format:	U2
Format:	U2			
25:24	<b>Pixel Hashing Table Entry y[2]x[12]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=12 and y=2	Format:	U2	
Format:	U2			
23:22	<b>Pixel Hashing Table Entry y[2]x[11]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=11 and y=2	Format:	U2	
Format:	U2			

## PIXEL\_HASH\_TABLE\_2BIT\_128ENTRY

	21:20	<b>Pixel Hashing Table Entry y[2]x[10]</b>
		Format: U2
		Indicates the pixelhash_id for the pixel block that has x=10 and y=2
	19:18	<b>Pixel Hashing Table Entry y[2]x[9]</b>
		Format: U2
		Indicates the pixelhash_id for the pixel block that has x=9 and y=2
	17:16	<b>Pixel Hashing Table Entry y[2]x[8]</b>
		Format: U2
		Indicates the pixelhash_id for the pixel block that has x=8 and y=2
	15:14	<b>Pixel Hashing Table Entry y[2]x[7]</b>
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=7 and y=2	
13:12	<b>Pixel Hashing Table Entry y[2]x[6]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=6 and y=2	
11:10	<b>Pixel Hashing Table Entry y[2]x[5]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=5 and y=2	
9:8	<b>Pixel Hashing Table Entry y[2]x[4]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=4 and y=2	
7:6	<b>Pixel Hashing Table Entry y[2]x[3]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=3 and y=2	
5:4	<b>Pixel Hashing Table Entry y[2]x[2]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=2 and y=2	
3:2	<b>Pixel Hashing Table Entry y[2]x[1]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=1 and y=2	



<b>PIXEL_HASH_TABLE_2BIT_128ENTRY</b>				
	1:0	<b>Pixel Hashing Table Entry y[2]x[0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=0 and y=2	Format:	U2
Format:	U2			
3	31:30	<b>Pixel Hashing Table Entry y[3]x[15]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=15 and y=3	Format:	U2
	Format:	U2		
	29:28	<b>Pixel Hashing Table Entry y[3]x[14]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=14 and y=3	Format:	U2
	Format:	U2		
	27:26	<b>Pixel Hashing Table Entry y[3]x[13]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=13 and y=3	Format:	U2
	Format:	U2		
	25:24	<b>Pixel Hashing Table Entry y[3]x[12]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=12 and y=3	Format:	U2
	Format:	U2		
	23:22	<b>Pixel Hashing Table Entry y[3]x[11]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=11 and y=3	Format:	U2
Format:	U2			
21:20	<b>Pixel Hashing Table Entry y[3]x[10]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=10 and y=3	Format:	U2	
Format:	U2			
19:18	<b>Pixel Hashing Table Entry y[3]x[9]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=9 and y=3	Format:	U2	
Format:	U2			
17:16	<b>Pixel Hashing Table Entry y[3]x[8]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=8 and y=3	Format:	U2	
Format:	U2			
15:14	<b>Pixel Hashing Table Entry y[3]x[7]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=7 and y=3	Format:	U2	
Format:	U2			

<b>PIXEL_HASH_TABLE_2BIT_128ENTRY</b>				
	13:12	<b>Pixel Hashing Table Entry y[3]x[6]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=6 and y=3		U2
		U2		
	11:10	<b>Pixel Hashing Table Entry y[3]x[5]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=5 and y=3		U2
		U2		
	9:8	<b>Pixel Hashing Table Entry y[3]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=3		U2
		U2		
	7:6	<b>Pixel Hashing Table Entry y[3]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=3		U2
	U2			
5:4	<b>Pixel Hashing Table Entry y[3]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=3		U2	
	U2			
3:2	<b>Pixel Hashing Table Entry y[3]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=3		U2	
	U2			
1:0	<b>Pixel Hashing Table Entry y[3]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=3		U2	
	U2			
4	31:30	<b>Pixel Hashing Table Entry y[4]x[15]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=15 and y=4		U2
		U2		
	29:28	<b>Pixel Hashing Table Entry y[4]x[14]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=14 and y=4		U2
	U2			
27:26	<b>Pixel Hashing Table Entry y[4]x[13]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=13 and y=4		U2	
	U2			

<b>PIXEL_HASH_TABLE_2BIT_128ENTRY</b>			
25:24	<b>Pixel Hashing Table Entry y[4]x[12]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=12 and y=4	Format:	U2
Format:	U2		
23:22	<b>Pixel Hashing Table Entry y[4]x[11]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=11 and y=4	Format:	U2
Format:	U2		
21:20	<b>Pixel Hashing Table Entry y[4]x[10]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=10 and y=4	Format:	U2
Format:	U2		
19:18	<b>Pixel Hashing Table Entry y[4]x[9]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=9 and y=4	Format:	U2
Format:	U2		
17:16	<b>Pixel Hashing Table Entry y[4]x[8]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=8 and y=4	Format:	U2
Format:	U2		
15:14	<b>Pixel Hashing Table Entry y[4]x[7]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=7 and y=4	Format:	U2
Format:	U2		
13:12	<b>Pixel Hashing Table Entry y[4]x[6]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=6 and y=4	Format:	U2
Format:	U2		
11:10	<b>Pixel Hashing Table Entry y[4]x[5]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=5 and y=4	Format:	U2
Format:	U2		
9:8	<b>Pixel Hashing Table Entry y[4]x[4]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=4 and y=4	Format:	U2
Format:	U2		
7:6	<b>Pixel Hashing Table Entry y[4]x[3]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=3 and y=4	Format:	U2
Format:	U2		

<b>PIXEL_HASH_TABLE_2BIT_128ENTRY</b>				
	5:4	<b>Pixel Hashing Table Entry y[4]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=4	Format:	U2
	Format:	U2		
	3:2	<b>Pixel Hashing Table Entry y[4]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=4	Format:	U2
Format:	U2			
1:0	<b>Pixel Hashing Table Entry y[4]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=4	Format:	U2	
Format:	U2			
5	31:30	<b>Pixel Hashing Table Entry y[5]x[15]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=15 and y=5	Format:	U2
	Format:	U2		
	29:28	<b>Pixel Hashing Table Entry y[5]x[14]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=14 and y=5	Format:	U2
	Format:	U2		
	27:26	<b>Pixel Hashing Table Entry y[5]x[13]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=13 and y=5	Format:	U2
	Format:	U2		
	25:24	<b>Pixel Hashing Table Entry y[5]x[12]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=12 and y=5	Format:	U2
Format:	U2			
23:22	<b>Pixel Hashing Table Entry y[5]x[11]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=11 and y=5	Format:	U2	
Format:	U2			
21:20	<b>Pixel Hashing Table Entry y[5]x[10]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=10 and y=5	Format:	U2	
Format:	U2			
19:18	<b>Pixel Hashing Table Entry y[5]x[9]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=9 and y=5	Format:	U2	
Format:	U2			

<b>PIXEL_HASH_TABLE_2BIT_128ENTRY</b>				
	17:16	<b>Pixel Hashing Table Entry y[5]x[8]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=8 and y=5	Format:	U2
	Format:	U2		
	15:14	<b>Pixel Hashing Table Entry y[5]x[7]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=7 and y=5	Format:	U2
	Format:	U2		
	13:12	<b>Pixel Hashing Table Entry y[5]x[6]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=6 and y=5	Format:	U2
	Format:	U2		
	11:10	<b>Pixel Hashing Table Entry y[5]x[5]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=5 and y=5	Format:	U2
	Format:	U2		
	9:8	<b>Pixel Hashing Table Entry y[5]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=5	Format:	U2
Format:	U2			
7:6	<b>Pixel Hashing Table Entry y[5]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=5	Format:	U2	
Format:	U2			
5:4	<b>Pixel Hashing Table Entry y[5]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=5	Format:	U2	
Format:	U2			
3:2	<b>Pixel Hashing Table Entry y[5]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=5	Format:	U2	
Format:	U2			
1:0	<b>Pixel Hashing Table Entry y[5]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=5	Format:	U2	
Format:	U2			
6	31:30	<b>Pixel Hashing Table Entry y[6]x[15]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 150px;">Format:</td><td>U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=15 and y=6	Format:	U2
Format:	U2			

## PIXEL\_HASH\_TABLE\_2BIT\_128ENTRY

	29:28	<b>Pixel Hashing Table Entry y[6]x[14]</b>
		Format: U2
		Indicates the pixelhash_id for the pixel block that has x=14 and y=6
	27:26	<b>Pixel Hashing Table Entry y[6]x[13]</b>
		Format: U2
		Indicates the pixelhash_id for the pixel block that has x=13 and y=6
	25:24	<b>Pixel Hashing Table Entry y[6]x[12]</b>
		Format: U2
		Indicates the pixelhash_id for the pixel block that has x=12 and y=6
	23:22	<b>Pixel Hashing Table Entry y[6]x[11]</b>
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=11 and y=6	
21:20	<b>Pixel Hashing Table Entry y[6]x[10]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=10 and y=6	
19:18	<b>Pixel Hashing Table Entry y[6]x[9]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=9 and y=6	
17:16	<b>Pixel Hashing Table Entry y[6]x[8]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=8 and y=6	
15:14	<b>Pixel Hashing Table Entry y[6]x[7]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=7 and y=6	
13:12	<b>Pixel Hashing Table Entry y[6]x[6]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=6 and y=6	
11:10	<b>Pixel Hashing Table Entry y[6]x[5]</b>	
	Format: U2	
	Indicates the pixelhash_id for the pixel block that has x=5 and y=6	

<b>PIXEL_HASH_TABLE_2BIT_128ENTRY</b>				
	9:8	<b>Pixel Hashing Table Entry y[6]x[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=4 and y=6		U2
		U2		
	7:6	<b>Pixel Hashing Table Entry y[6]x[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=3 and y=6		U2
		U2		
	5:4	<b>Pixel Hashing Table Entry y[6]x[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=2 and y=6		U2
	U2			
3:2	<b>Pixel Hashing Table Entry y[6]x[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=1 and y=6		U2	
	U2			
1:0	<b>Pixel Hashing Table Entry y[6]x[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=0 and y=6		U2	
	U2			
7	31:30	<b>Pixel Hashing Table Entry y[7]x[15]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=15 and y=7		U2
		U2		
	29:28	<b>Pixel Hashing Table Entry y[7]x[14]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=14 and y=7		U2
		U2		
	27:26	<b>Pixel Hashing Table Entry y[7]x[13]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=13 and y=7		U2
	U2			
25:24	<b>Pixel Hashing Table Entry y[7]x[12]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=12 and y=7		U2	
	U2			
23:22	<b>Pixel Hashing Table Entry y[7]x[11]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;"> </td><td style="width: 50px;">U2</td></tr></table> Indicates the pixelhash_id for the pixel block that has x=11 and y=7		U2	
	U2			

## PIXEL\_HASH\_TABLE\_2BIT\_128ENTRY

21:20	<p><b>Pixel Hashing Table Entry y[7]x[10]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=10 and y=7</p>	Format:	U2
Format:	U2		
19:18	<p><b>Pixel Hashing Table Entry y[7]x[9]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=9 and y=7</p>	Format:	U2
Format:	U2		
17:16	<p><b>Pixel Hashing Table Entry y[7]x[8]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=8 and y=7</p>	Format:	U2
Format:	U2		
15:14	<p><b>Pixel Hashing Table Entry y[7]x[7]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=7 and y=7</p>	Format:	U2
Format:	U2		
13:12	<p><b>Pixel Hashing Table Entry y[7]x[6]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=6 and y=7</p>	Format:	U2
Format:	U2		
11:10	<p><b>Pixel Hashing Table Entry y[7]x[5]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=5 and y=7</p>	Format:	U2
Format:	U2		
9:8	<p><b>Pixel Hashing Table Entry y[7]x[4]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=4 and y=7</p>	Format:	U2
Format:	U2		
7:6	<p><b>Pixel Hashing Table Entry y[7]x[3]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=3 and y=7</p>	Format:	U2
Format:	U2		
5:4	<p><b>Pixel Hashing Table Entry y[7]x[2]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=2 and y=7</p>	Format:	U2
Format:	U2		
3:2	<p><b>Pixel Hashing Table Entry y[7]x[1]</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> </table> <p>Indicates the pixelhash_id for the pixel block that has x=1 and y=7</p>	Format:	U2
Format:	U2		



<b>PIXEL_HASH_TABLE_2BIT_128ENTRY</b>				
	1:0	<b>Pixel Hashing Table Entry y[7]x[0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="width: 20%; text-align: center;">U2</td> </tr> </table> Indicates the pixelhash_id for the pixel block that has x=0 and y=7	Format:	U2
Format:	U2			

## Pixel Sample Mask Render Target Message Header Control

MHC_RT_PSM - Pixel Sample Mask Render Target Message Header Control						
Size (in bits):		32				
Default Value:		0x00000000				
DWord	Bit	Description				
0	31:16	<p><b>Dispatched Pixel/Sample Enables</b></p> <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> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>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.</p>	Format:	U16	<b>Programming Notes</b>	
	Format:	U16				
<b>Programming Notes</b>						
15:0	<p><b>Pixel/Sample Enables</b></p> <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> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>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.</p>	Format:	U16	<b>Programming Notes</b>		
Format:	U16					
<b>Programming Notes</b>						









PPHWSP_LAYOUT - PPHWSP_LAYOUT											
4	31:0	<b>Ring Head Pointer Storage</b> <table border="1"> <thead> <tr> <th colspan="2">Description</th> </tr> </thead> <tbody> <tr> <td colspan="2">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).</td> </tr> </tbody> </table>	Description		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).						
Description											
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	351:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO										
Format:	MBZ										
16	0	<b>Cumulative Context Run Time</b> <p>This has the cumulative run time of the context on HW. HW reports CTX_TIMESTAMP to this location on a context switch.</p> <p>This value is written after the context save is complete. The value that is saved in the context image does not include the time between the saving of the cumulative value to context to the time we complete the save. If required for the value to always increment and not take the context save into consideration, driver must look at the value in the context image.</p>									
17	31:1	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO										
Format:	MBZ										
	0	<b>Element Switch</b> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Indicates the context is not submitted as the first element in the execlist.</td> </tr> <tr> <td>1</td> <td></td> <td>Indicates the corresponding context has been submitted as first element of the execlist. Preempt Request Received Timestamp is the time when the pending execlist has been submitted to HW. Note that across multiple submissions a given context could be first or second element of an execlist. This bit will get set if the context has been submitted as the first element in the execlist.</td> </tr> </tbody> </table>	Value	Name	Description	0		Indicates the context is not submitted as the first element in the execlist.	1		Indicates the corresponding context has been submitted as first element of the execlist. Preempt Request Received Timestamp is the time when the pending execlist has been submitted to HW. Note that across multiple submissions a given context could be first or second element of an execlist. This bit will get set if the context has been submitted as the first element in the execlist.
Value	Name	Description									
0		Indicates the context is not submitted as the first element in the execlist.									
1		Indicates the corresponding context has been submitted as first element of the execlist. Preempt Request Received Timestamp is the time when the pending execlist has been submitted to HW. Note that across multiple submissions a given context could be first or second element of an execlist. This bit will get set if the context has been submitted as the first element in the execlist.									
18..19	63:0	<b>Preempt Request Received Timestamp</b> TIMESTAMP register sampled on preemption request is reported.									
20..21	63:0	<b>Context Restore Complete Timestamp</b> TIMESTAMP register sampled on context restore complete is reported.									
22..23	63:0	<b>Context Save Finished Timestamp</b> TIMESTAMP register sampled on context save completion is reported.									
24..27	127:0	<b>MI_SEMAPHORE_WAIT</b> MI_SEMAPHORE_WAIT command on which the context got switched out due to semaphore wait. This field is only valid and must be looked at when the context switch reason in context status buffer is stated as Wait on Semaphore.									

PPHWSP_LAYOUT - PPHWSP_LAYOUT						
28..31	127:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					
32..33 This field describes the most recent context switch status of the corresponding context.	63:0	<b>Context Switch Status Qword</b>				
34..1020	31583:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					



## Predicate Barrier Message Data Payload

<b>MDP_PREDICATE_BARRIER - Predicate Barrier Message Data Payload</b>						
Source:	EuSubFunctionGateway					
Size (in bits):	256					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000					
<p>This response message is sent back only if the Gateway Barrier Message specifies that this is a predicated barrier. This response is written to the GRF writeback location, and the response length specified in the send message to the EU must be 1.</p>						
DWord	Bit	Description				
0	31:16	<b>Reserved</b>				
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
15:0	<b>Predicated Barrier Mask Sum</b>					
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> <p>This field is a sum of the predicate mask bits sent by each thread. This field (and the DW containing it) is not written if the barrier is not marked as a predicated barrier. The kernel should compare this field to 0 for the predicated OR function and compare it to the workgroup size for the predicated AND function.</p>	Format:	U16			
Format:	U16					
1..7	223:0	<b>Reserved</b>				
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
		Access:	RO			
Format:	MBZ					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
Access:	RO					
Format:	MBZ					



## Qword Data Payload Register

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





## Qword SIMD8 Atomic Operation Return Data Message Data Payload

MDP_AOP8_QW1 - Qword SIMD8 Atomic Operation Return Data Message Data Payload		
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	<b>Slot[7:0] Qword[31:0]</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MDCR_DW</b></td> </tr> </table> <p>Specifies the lower 32-bits of Slot [7:0] Return data</p>
Format:	<b>MDCR_DW</b>	
1.0-1.7	255:0	<b>Slot[7:0] Qword[63:32]</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MDCR_DW</b></td> </tr> </table> <p>Specifies the upper 32-bits of Slot [7:0] Return data</p>
Format:	<b>MDCR_DW</b>	



## Qword SIMD8 Data Payload

<b>MDP_QW_SIMD8 - Qword SIMD8 Data Payload</b>		
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	<b>Data[3:0]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDCR_QW</span> Specifies the Slot [3:0] data
1.0-1.7	255:0	<b>Data[7:4]</b> Format: <span style="border: 1px solid black; padding: 2px;">MDCR_QW</span> Specifies the Slot [7:4] data



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



## Qword SIMD16 Atomic Operation Return Data Message Data Payload

MDP_AOP16_QW1 - Qword SIMD16 Atomic Operation Return Data Message Data Payload		
Size (in bits):	1024	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Slot[7:0] Qword[31:0]</b>
		Format: <b>MDCR_DW</b> Specifies the lower 32-bits of Return data for Slot [7:0]
1.0-1.7	255:0	<b>Slot[15:8] Qword[31:0]</b>
		Format: <b>MDCR_DW</b> Specifies the lower 32-bits of Return data for Slot [15:8]
2.0-2.7	255:0	<b>Slot[7:0] Qword[63:32]</b>
		Format: <b>MDCR_DW</b> Specifies the upper 32-bits of Return data for Slot [7:0]
3.0-3.7	255:0	<b>Slot[15:8] Qword[63:32]</b>
		Format: <b>MDCR_DW</b> Specifies the upper 32-bits of Return data for Slot [15:8]



## Read-Only Data Port Message Types

MT_DP_RO - Read-Only Data Port Message Types																										
Source:	EuSubFunctionReadOnlyDataPort																									
Size (in bits):	5																									
Default Value:	0x00000000																									
Description																										
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.																										
Lists all the Message Types in a Read-Only Data Port Message Descriptor [18:14]. Read operations from the Constant 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	4:0	<p><b>Message Type</b> Specifies type of message</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>MT_CC_OWB <b>[Default]</b></td> <td>Oword Block Read Constant Cache message</td> </tr> <tr> <td>01h</td> <td>MT_CC_OWUB</td> <td>Unaligned Oword Block Read Constant Cache message</td> </tr> <tr> <td>03h</td> <td>MT_CC_DWS</td> <td>Dword Scattered Read Constant Cache message</td> </tr> <tr> <td>04h</td> <td>MT_SC_OWUB</td> <td>Unaligned Oword Block Read Sampler Cache message</td> </tr> <tr> <td>05h</td> <td>MT_SC_MB</td> <td>Media Block Read Sampler Cache message</td> </tr> <tr> <td>06h</td> <td>MT_RSI</td> <td>Read Surface Info message</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Value	Name	Description	00h	MT_CC_OWB <b>[Default]</b>	Oword Block Read Constant Cache message	01h	MT_CC_OWUB	Unaligned Oword 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
Value	Name	Description																								
00h	MT_CC_OWB <b>[Default]</b>	Oword Block Read Constant Cache message																								
01h	MT_CC_OWUB	Unaligned Oword 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		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0	31:0	<b>U</b> Format: U32 Specifies the U channel address offset.
0.1	31:0	<b>V</b> Format: U32 Specifies the V channel address offset.
0.2	31:0	<b>R</b> Format: U32 Specifies the R channel address offset.
0.3	31:0	<b>LOD</b> Format: <b>MACD_LOD</b> Specifies the LOD.
0.4-0.7	127:0	<b>Reserved</b> Access: RO Format: MBZ

## Read Surface Info Address Payload

<b>DP_ASTATE_INFO_PAYLOAD - Read Surface Info Address Payload</b>				
Size (in bits):	128			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000			
<b>DWord</b>	<b>Bit</b>	<b>Description</b>		
0.0	31:0	<p><b>U</b></p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the U channel address offset.</p>	Format:	U32
Format:	U32			
0.1	31:0	<p><b>V</b></p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the V channel address offset.</p>	Format:	U32
Format:	U32			
0.2	31:0	<p><b>R</b></p> <table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> </table> <p>Specifies the R channel address offset.</p>	Format:	U32
Format:	U32			
0.3	31:0	<p><b>LOD</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>MACD_LOD</b></td> </tr> </table> <p>Specifies the LOD.</p>	Format:	<b>MACD_LOD</b>
Format:	<b>MACD_LOD</b>			

## Read Surface Info Data Payload

<b>MDP_RSI - Read Surface Info Data Payload</b>		
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	<b>Reserved</b>
		Access: RO
		Format: MBZ
0.6-0.7	63:0	<b>Instruction Base Address</b>
		Format: GraphicsAddress[63:0]
		Instruction Base Address from STATE_BASE_ADDRESS, extended to 64-bit format.
		<b>Programming Notes</b>
		The 48-bit address is returned in a 64-bit address in canonical form.
		<b>Restriction</b>
		The Instruction Base Address returned is incorrect if the thread is from CCS queue.
1.0	31:0	<b>Width</b>
		Format: U32
		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.
1.1	31:0	<b>Height</b>
		Format: U32
		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.
1.2	31:0	<b>Depth</b>
		Format: U32
		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.
1.3	31:0	<b>MIP Count</b>
		Format: U32
		MIP Count from RENDER_SURFACE_STATE, range [0, 14], zero extended to 32 bits.

<b>MDP_RSI - Read Surface Info Data Payload</b>																										
1.4	31:0	<b>Surface Type</b>																								
		Format: U32																								
		Surface Type from RENDER_SURFACE_STATE, zero extended to 32 bits																								
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>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>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	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																								
7h	SURTYPE_NULL	Null surface																								
Others	Reserved	Reserved																								
1.5	31:0	<b>Surface Format</b>																								
		Format: U32																								
		Surface Format from RENDER_SURFACE_STATE (U9), zero extended to 32 bits.																								
1.6-1.7	63:0	<b>Reserved</b>																								
		Access: RO																								
		Format: MBZ																								

## RENDER\_SURFACE\_STATE

<b>RENDER_SURFACE_STATE</b>																										
Exists If:	//[MessageType] != 'Sample_8x8'																									
Size (in bits):	512																									
Default Value:	0x0001C000, 0x80000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000																									
This is the normal surface state used by all messages that use SURFACE_STATE except those that use MEDIA_SURFACE_STATE.																										
DWord	Bit	Description																								
0	31:2 9	<p><b>Surface Type</b> This field defines the type of 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;">0h</td> <td style="text-align: center;">SURFTYPE_1D</td> <td>Defines a 1-dimensional map or array of maps</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">SURFTYPE_2D</td> <td>Defines a 2-dimensional map or array of maps</td> </tr> <tr> <td style="text-align: center;">2h</td> <td style="text-align: center;">SURFTYPE_3D</td> <td>Defines a 3-dimensional (volumetric) map</td> </tr> <tr> <td style="text-align: center;">3h</td> <td style="text-align: center;">SURFTYPE_CUBE</td> <td>Defines a cube map or array of cube maps</td> </tr> <tr> <td style="text-align: center;">4h</td> <td style="text-align: center;">SURFTYPE_BUFFER</td> <td>Defines an element in a buffer</td> </tr> <tr> <td style="text-align: center;">6h</td> <td style="text-align: center;">Reserved</td> <td></td> </tr> <tr> <td style="text-align: center;">7h</td> <td style="text-align: center;">SURFTYPE_NULL</td> <td>Defines a null surface</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>A null surface is used in instances where an actual surface is not bound. When a write message is generated to a null surface, no actual surface is written to. When a read message (including any sampling engine message) is generated to a null surface, the result is all zeros. Note that a null surface type is allowed to be used with all messages, even if it is not specifically indicated as supported. All of the remaining fields in surface state are ignored for null surfaces, with the following exceptions:</p> <ul style="list-style-type: none"> <li>• <b>Width, Height, Depth, LOD, Render Target View Extent, Minimum Array element</b> fields must match for all surfaces that compose the current render target.</li> </ul> <p>All sampling engine and data port messages support null surfaces with the above behavior, even if not mentioned as specifically supported, except for the following:</p> <ul style="list-style-type: none"> <li>• The <b>Surface Type</b> of a surface used as a render target (accessed via the Data Port's Render Target Write message) must be the same as the <b>Surface Type</b> of all other render targets and of the depth / stencil buffer (defined in 3DSTATE_DEPTH_BUFFER, 3DSTATE_STENCIL_BUFFER), unless either the stencil buffer, depth buffer or render targets are SURFTYPE_NULL.</li> </ul>	Value	Name	Description	0h	SURFTYPE_1D	Defines a 1-dimensional map or array of maps	1h	SURFTYPE_2D	Defines a 2-dimensional map or array of maps	2h	SURFTYPE_3D	Defines a 3-dimensional (volumetric) map	3h	SURFTYPE_CUBE	Defines a cube map or array of cube maps	4h	SURFTYPE_BUFFER	Defines an element in a buffer	6h	Reserved		7h	SURFTYPE_NULL	Defines a null surface
Value	Name	Description																								
0h	SURFTYPE_1D	Defines a 1-dimensional map or array of maps																								
1h	SURFTYPE_2D	Defines a 2-dimensional map or array of maps																								
2h	SURFTYPE_3D	Defines a 3-dimensional (volumetric) map																								
3h	SURFTYPE_CUBE	Defines a cube map or array of cube maps																								
4h	SURFTYPE_BUFFER	Defines an element in a buffer																								
6h	Reserved																									
7h	SURFTYPE_NULL	Defines a null surface																								



<b>RENDER_SURFACE_STATE</b>					
28	<p><b>Surface Array</b></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> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </table> <p>If this field is <i>enabled</i>, the <b>Surface Type</b> must be SURFTYPE_1D, SURFTYPE_2D, or SURFTYPE_CUBE.            If this field is <i>disabled</i> and <b>Surface Type</b> is SURFTYPE_1D, SURFTYPE_2D, or SURFTYPE_CUBE, the <b>Depth</b> field must be set to zero.</p> <p>To avoid performance issues with the sampler texture cache this bit should not be set unless the depth of the arrayed surface is greater than 1 (Depth field &gt;1).</p>	Format:	Enable	Programming Notes	
Format:	Enable				
Programming Notes					
27	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				
26:1 8	<p><b>Surface Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td><b>SURFACE_FORMAT</b></td> </tr> </table> <p>This field specifies the format of the surface or element within this surface. This field is ignored for all data port messages other than the render target message and streamed vertex buffer write message. Some forms of the media block messages use the surface format.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> </table> <p>If <b>Number of Multisamples</b> 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"> <li>• Any compressed texture format (BC*, DXT*, FXT*, ETC*, EAC*)</li> <li>• Any YCRCB* format</li> </ul> <p>If <b>ASTC_Enable</b> is set to 1:</p> <ul style="list-style-type: none"> <li>• ASTC_Profile: ASTC_Profile can be freely set to 1 or 0.</li> <li>• ASTC_BlockDimension: Bit [25] will be set to 0 if ASTC_Profile is LDR-profile since only 2D blocks are supported in LDR profile. 2D HDR-profile can be supported where the ASTC_Profile is set at 1 (Full-Profile) and ASTC_BlockDimension is set as 0 (2D).</li> </ul> <p>This field cannot be a YUV (YCRCB*) or compressed (BC*, DXT*, FXT*, ETC*, EAC*) format if the <b>Surface Type</b> is SURFTYPE_BUFFER.            This field cannot be a compressed (BC*, DXT*, FXT*, ETC*, EAC*) format if the <b>Surface Type</b> is SURFTYPE_1D.</p>	Format:	<b>SURFACE_FORMAT</b>	Programming Notes	
Format:	<b>SURFACE_FORMAT</b>				
Programming Notes					
17:1 6	<p><b>Surface Vertical Alignment</b></p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Description</th> </tr> </table> <p><b>For Sampling Engine and Render Target Surfaces:</b> 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>	Description			
Description					

## RENDER\_SURFACE\_STATE

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.

The vertical alignment field (VALIGN) is also used to determine the start of the surface in memory. All surface are vertically aligned to VALIGN rows or VALIGN rows times the block height for block-compressed surface formats.

**For other surfaces:** This field is ignored.

Value	Name	Description
0h	Reserved	Reserved
1h	VALIGN_4 <b>[Default]</b>	Vertical alignment factor j = 4
2h	VALIGN_8	Vertical alignment factor j = 8
3h	VALIGN_16	Vertical alignment factor j = 16

### Programming Notes

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.

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.

This field should also be set to VALIGN\_8 if the surface was rendered as a D16\_UNORM depth buffer, for render target of 1X/4X/16X.

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.

See **Surface Layout and Tiling** for a full description of how i and j parameters are used to determine horizontal and vertical offset to the start of a MIP.

15:1  
4 **Surface Horizontal Alignment**

### Description

For Sampling Engine and Render Target Surfaces: This field specifies the horizontal alignment requirement for the surface.

This field is used for alignment when LOD >= Mip Tail Start LOD  
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.

**For other surfaces:** This field is ignored.

## 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 <b>[Default]</b>	Horizontal alignment factor j = 16

### Programming Notes

See **Alignment Unit Size** For restrictions on Horizontal Alignment.

When Auxiliary Surface Mode is set to AUX\_MCS\_LCE or AUX\_CCS\_E, HALIGN\_16 must be used unless contradicts **Alignment Unit Size** restrictions.

13:1  
2

### Tile Mode

#### Description

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.

This field specifies the type of memory tiling (Linear, XMajor, or YMajor) employed to tile this surface. See *Memory Interface Functions* for details on memory tiling and restrictions.

Value	Name	Description	Programming Notes
0h	LINEAR	Linear mode (no tiling)	
1h	Reserved	Reserved	
2h	XMAJOR	X major tiling	
3h	YMAJOR	Y major tiling	

### Programming Notes

- Refer to *Memory Data Formats* 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.
- 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\_1D this field must be TILEMODE\_LINEAR, unless **Sampler Legacy 1D Map Layout Disable** is set to 0, in which case TILEMODE\_YMAJOR is also allowed. **Tiled Resource Mode** must be set to TRMODE\_NONE for these cases.

If **Surface Format** is ASTC\*, this field must be TILEMODE\_YMAJOR.

## RENDER\_SURFACE\_STATE

11	<p><b>Vertical Line Stride</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">U1</td> </tr> </table> <p><b>For 2D Non-Array Surfaces accessed via the Sampling Engine or Data Cache Data Port:</b> Specifies number of lines (0 or 1) to skip between logically adjacent lines - provides support of interleaved (field) surfaces as textures.</p> <p><b>For Other Surfaces:</b> Vertical Line Stride must be zero.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This bit must not be set if the surface format is a compressed type (BCn*, FXT1, ETC*, EAC*).</p> <p>This bit must not be set if the <b>Auxiliary Surface Mode</b> is not AUX_NONE.</p> <p>If this bit is set on a sampling engine surface, the mip mode filter must be set to MIPFILTER_NONE</p>	Format:	U1							
Format:	U1									
10	<p><b>Vertical Line Stride Offset</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">U1</td> </tr> </table> <p><b>For 2D Non-Array Surfaces accessed via the Sampling Engine or Data Cache Data Port:</b> Specifies the offset of the initial line from the beginning of the buffer. Ignored when Vertical <b>Line Stride</b> is 0.</p> <p><b>For Other Surfaces:</b> Vertical Line Stride Offset must be zero.</p>	Format:	U1							
Format:	U1									
9	<p><b>Sampler L2 Out of Order Mode Disable</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">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 1:4, 1:2 in L1 or not expanded at all. This would include all lossless compressed cases.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This bit must be set for the following surface types: BC2_UNORM BC3_UNORM BC5_UNORM BC5_SNORM BC7_UNORM</p>	Format:	Disable							
Format:	Disable									
8	<p><b>Render Cache Read Write Mode</b></p> <p><b>For Surfaces accessed via the Data Port to Render Cache:</b> 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> <p><b>For Surfaces accessed via the Sampling Engine or Data Port to Texture Cache or Data Cache:</b> This field is reserved : MBZ</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 55%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>Write-Only Cache</td> <td>Allocating write-only cache for a write miss</td> </tr> <tr> <td style="text-align: center;">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;"><b>Programming Notes</b></p> <p>This field is provided for performance optimization for Render Cache read/write accesses.</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								

## RENDER\_SURFACE\_STATE

7:6 **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.  
**For Other Surfaces:** Reserved : MBZ

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 **Cube Face Enable - Negative X**

Format:	Enable
---------	--------

**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.

### Programming Notes

When TEXCOORDMODE\_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).

This field must be programmed to 1h (enabled) whenever **Surface Type** is programmed to **SURFTYPE\_CUBE**

4 **Cube Face Enable - Positive X**

Format:	Enable
---------	--------

**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.

### Programming Notes

When TEXCOORDMODE\_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).

This field must be programmed to 1h (enabled) whenever **Surface Type** is programmed to **SURFTYPE\_CUBE**

## RENDER\_SURFACE\_STATE

3	<p><b>Cube Face Enable - Negative Y</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p><b>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine:</b> 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> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td>When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> <tr> <td>This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b></td> </tr> </table>	Format:	Enable	<b>Programming Notes</b>	When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).	This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b>
Format:	Enable					
<b>Programming Notes</b>						
When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).						
This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b>						
2	<p><b>Cube Face Enable - Positive Y</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p><b>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine:</b> 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> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td>When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> <tr> <td>This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b></td> </tr> </table>	Format:	Enable	<b>Programming Notes</b>	When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).	This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b>
Format:	Enable					
<b>Programming Notes</b>						
When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).						
This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b>						
1	<p><b>Cube Face Enable - Negative Z</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p><b>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine:</b> 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> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td>When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> <tr> <td>This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b></td> </tr> </table>	Format:	Enable	<b>Programming Notes</b>	When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).	This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b>
Format:	Enable					
<b>Programming Notes</b>						
When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).						
This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b>						
0	<p><b>Cube Face Enable - Positive Z</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p><b>For SURFTYPE_CUBE Surfaces accessed via the Sampling Engine:</b> 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</p>	Format:	Enable			
Format:	Enable					

<b>RENDER_SURFACE_STATE</b>																																		
	<p>that storage for disabled faces must be provided.</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">When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).</td> </tr> <tr> <td colspan="2">This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b></td> </tr> </table>	Programming Notes		When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).		This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b>																												
Programming Notes																																		
When TEXCOORDMODE_CLAMP is used when accessing a cube map, this field must be programmed to 1b (face enabled).																																		
This field must be programmed to 1h (enabled) whenever <b>Surface Type</b> is programmed to <b>SURFTYPE_CUBE</b>																																		
1	<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">31</td> <td colspan="2"><b>Enable Unorm Path in Color Pipe</b></td> </tr> <tr> <td></td> <td style="width: 60%;">Format:</td> <td style="text-align: center;">Enable</td> </tr> <tr> <td colspan="3">Enables Unorm Path (fixed Point Conversion of floating point for fill and blend in DAPRSS) in color Pipe.</td> </tr> <tr> <td></td> <td style="text-align: center;"><b>Value</b></td> <td style="text-align: center;"><b>Name</b></td> </tr> <tr> <td></td> <td style="text-align: center;">1</td> <td style="text-align: center;">ENABLE <b>[Default]</b></td> </tr> <tr> <td></td> <td style="text-align: center;">0</td> <td style="text-align: center;">DISABLE</td> </tr> <tr> <td></td> <td></td> <td style="text-align: center;"><b>Description</b></td> </tr> <tr> <td></td> <td></td> <td style="text-align: center;">Enables Unorm Path in Color Pipe.</td> </tr> <tr> <td></td> <td></td> <td style="text-align: center;">Disables Unorm path in Color Pipe.</td> </tr> <tr> <td colspan="3" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="3">This bit should never be programmed to 0</td> </tr> </table>	31	<b>Enable Unorm Path in Color Pipe</b>			Format:	Enable	Enables Unorm Path (fixed Point Conversion of floating point for fill and blend in DAPRSS) in color Pipe.				<b>Value</b>	<b>Name</b>		1	ENABLE <b>[Default]</b>		0	DISABLE			<b>Description</b>			Enables Unorm Path in Color Pipe.			Disables Unorm path in Color Pipe.	<b>Programming Notes</b>			This bit should never be programmed to 0		
31	<b>Enable Unorm Path in Color Pipe</b>																																	
	Format:	Enable																																
Enables Unorm Path (fixed Point Conversion of floating point for fill and blend in DAPRSS) in color Pipe.																																		
	<b>Value</b>	<b>Name</b>																																
	1	ENABLE <b>[Default]</b>																																
	0	DISABLE																																
		<b>Description</b>																																
		Enables Unorm Path in Color Pipe.																																
		Disables Unorm path in Color Pipe.																																
<b>Programming Notes</b>																																		
This bit should never be programmed to 0																																		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">30:2</td> <td colspan="2"><b>Memory Object Control State</b></td> </tr> <tr> <td style="text-align: center;">4</td> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MEMORY_OBJECT_CONTROL_STATE</b></td> </tr> <tr> <td colspan="3">Specifies the memory object control state for this surface and the associated Auxiliary surface (if any).</td> </tr> </table>	30:2	<b>Memory Object Control State</b>		4	Format:	<b>MEMORY_OBJECT_CONTROL_STATE</b>	Specifies the memory object control state for this surface and the associated Auxiliary surface (if any).																										
30:2	<b>Memory Object Control State</b>																																	
4	Format:	<b>MEMORY_OBJECT_CONTROL_STATE</b>																																
Specifies the memory object control state for this surface and the associated Auxiliary surface (if any).																																		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 5%; text-align: center;">23:1</td> <td colspan="2"><b>Base Mip Level</b></td> </tr> <tr> <td style="text-align: center;">9</td> <td style="width: 60%;">Format:</td> <td style="text-align: center;">U4.1</td> </tr> <tr> <td colspan="3">Range: [0.0, 14.0]</td> </tr> <tr> <td colspan="3">Specifies which mip level is considered the "base" level when determining mag-vs-min filter and selecting the "base" mip level.</td> </tr> <tr> <td colspan="3" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="3">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>	23:1	<b>Base Mip Level</b>		9	Format:	U4.1	Range: [0.0, 14.0]			Specifies which mip level is considered the "base" level when determining mag-vs-min filter and selecting the "base" mip level.			<b>Programming Notes</b>			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.																	
23:1	<b>Base Mip Level</b>																																	
9	Format:	U4.1																																
Range: [0.0, 14.0]																																		
Specifies which mip level is considered the "base" level when determining mag-vs-min filter and selecting the "base" mip level.																																		
<b>Programming Notes</b>																																		
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: 5%; text-align: center;">18</td> <td colspan="2"><b>Corner Texel Mode</b></td> </tr> <tr> <td></td> <td style="width: 60%;">Format:</td> <td style="text-align: center;">Enable</td> </tr> <tr> <td colspan="3">This field, when ENABLED, indicates when a surface is using corner texel-mode for sampling. Corner Texel Mode is ignored for Planar YUV/YCrCb surface formats. Corner Texel Mode is ignored for sample_8X8 and sample_unorm message types. Corner Texel Mode is not supported with Non-Normalized Coordinates Does not support legacy sampler features set0 See legacy sampler page for more details</td> </tr> </table>	18	<b>Corner Texel Mode</b>			Format:	Enable	This field, when ENABLED, indicates when a surface is using corner texel-mode for sampling. Corner Texel Mode is ignored for Planar YUV/YCrCb surface formats. Corner Texel Mode is ignored for sample_8X8 and sample_unorm message types. Corner Texel Mode is not supported with Non-Normalized Coordinates Does not support legacy sampler features set0 See legacy sampler page for more details																										
18	<b>Corner Texel Mode</b>																																	
	Format:	Enable																																
This field, when ENABLED, indicates when a surface is using corner texel-mode for sampling. Corner Texel Mode is ignored for Planar YUV/YCrCb surface formats. Corner Texel Mode is ignored for sample_8X8 and sample_unorm message types. Corner Texel Mode is not supported with Non-Normalized Coordinates Does not support legacy sampler features set0 See legacy sampler page for more details																																		

<b>RENDER_SURFACE_STATE</b>			
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	Disable <b>[Default]</b>	When programmed to 0h, Corner Texel Mode is disabled. This means texel references are shifted a half-texel from the upper-right corner of the texture map which is the standard texel reference mode.
	1h	Enable	When programmed to 1h, Corner Texel Mode is enabled. The location of a sampled texel on a texture map is shifted a half-texel to the upper-left, meaning texel (0,0) is in the exact upper-left corner of the surface.
17	<b>Double Fetch Disable</b>		
	Format:		Disable
	This bit is intended to disable the "double fetch" of adjacent cache-lines in most all cases. Double fetch is a performance mode, but for some surface types for formats it may be lower performance due to fetching unused cache-lines.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	Enable <b>[Default]</b>	When programmed to 0h, double fetches are allowed.
	1h	Disable	When programmed to 1h, double fetches are disabled.
16	<b>Reserved</b>		
	Access:		RO
	Format:		MBZ
15	<b>Sample Tap Discard Disable</b>		
	This bit forces sample tap discard filter mode to be disabled for this surface state. This bit must be set for surfaces which are no Alpha Channel such as R8G8B8_UNORM.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	DISABLE <b>[Default]</b>	When programmed to 0h, Sample Tap Discard filter mode is allowed and is not disabled by this bit. This bit is ignored if Sample Tap Discard is not enabled in the Sampler State.
	1h	ENABLE	When programmed to 1h, Sample Tap Discard filter mode will be disabled even if enabled through Sampler State
	<b>Programming Notes</b>		
	This bit must be set for all Planar YUV surface formats (e.g. PLANAR_420_8, PLANAR_420_16)		
14:0	<b>Surface QPitch</b>		
	Format:		U17[16:2]
	The interpretation of this field is dependent on Surface Type as follows:		
	<ul style="list-style-type: none"> <li>• SURFTYPE_1D: distance in <i>pixels</i> between array slices</li> <li>• 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.</li> <li>• SURFTYPE_3D: distance in <i>rows</i> between R-slices [<b>Note:</b> 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.</li> </ul>		



<b>RENDER_SURFACE_STATE</b>																					
		<ul style="list-style-type: none"> <li>Other surface types: field is ignored</li> </ul> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,32767]</td> <td></td> <td>1 is added to the value and it is multiplied by 4 to determine the height in rows.</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p><b>For Surface Type 1D:</b> This field must be set to an integer multiple of the <b>Surface Horizontal Alignment</b></p> <p><b>For Surface Type 2D, CUBE:</b> This field must be set to an integer multiple of the <b>Surface Vertical Alignment</b></p> <p><b>For Surface Type 3D: Tile Mode != Linear:</b> This field must be set to an integer multiple of the tile height (<math>2^{Cv}</math>) <b>Tile Mode == Linear:</b> This field must be set to an integer multiple of the Surface Vertical Alignment</p> <p><b>Note:</b> for compressed textures (BC*, FXT1, ETC*, EAC*), this field is in units of rows of compression blocks.</p> <p>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.</p>		Value	Name	Description	[0,32767]		1 is added to the value and it is multiplied by 4 to determine the height in rows.												
Value	Name	Description																			
[0,32767]		1 is added to the value and it is multiplied by 4 to determine the height in rows.																			
2	31	<p><b>Depth/Stencil Resource</b></p> <p style="text-align: center;"><b>Description</b></p> <p>This bit field, when set, indicates if the resource is created as Depth/Stencil resource.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>SW must set this bit for any resource that was created with Depth/Stencil resource flag. Setting this bit allows HW to properly interpret the data-layout for various cases. For any resource that's created without Depth/Stencil resource flag, it must be reset.</p>																			
	30	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ														
Access:	RO																				
Format:	MBZ																				
	29:1	<p><b>Height</b></p> <table border="1"> <tr> <td>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"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> <th>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>[0,8191]</td> <td></td> <td>height of surface - 1 (y/v dimension) If Vertical Line Stride is set for a 2D surface, then the maximum allowed height is <math>2^8</math>.</td> <td>([SurfaceType] == 'SURFTYPE_2D') &amp;&amp; ([VerticalLineStride] == 1)</td> </tr> <tr> <td>[0,16383]</td> <td></td> <td>height of surface - 1 (y/v</td> <td>([SurfaceType] == 'SURFTYPE_2D') &amp;&amp;</td> </tr> </tbody> </table>		Format:	U14-1	Value	Name	Description	Exists If	[0,0]		must be zero	[Surface Type] == 'SURFTYPE_1D'	[0,8191]		height of surface - 1 (y/v dimension) If Vertical Line Stride is set for a 2D surface, then the maximum allowed height is $2^8$ .	([SurfaceType] == 'SURFTYPE_2D') && ([VerticalLineStride] == 1)	[0,16383]		height of surface - 1 (y/v	([SurfaceType] == 'SURFTYPE_2D') &&
Format:	U14-1																				
Value	Name	Description	Exists If																		
[0,0]		must be zero	[Surface Type] == 'SURFTYPE_1D'																		
[0,8191]		height of surface - 1 (y/v dimension) If Vertical Line Stride is set for a 2D surface, then the maximum allowed height is $2^8$ .	([SurfaceType] == 'SURFTYPE_2D') && ([VerticalLineStride] == 1)																		
[0,16383]		height of surface - 1 (y/v	([SurfaceType] == 'SURFTYPE_2D') &&																		
	6																				

<b>RENDER_SURFACE_STATE</b>			
		dimension)	([VerticalLineStride] == 0)
[0,2047]		height of surface - 1 (y/v)	[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'
<b>Programming Notes</b>			
For typed buffer and structured buffer surfaces, the number of entries in the buffer ranges from 1 to 2 <sup>27</sup> . For raw buffer surfaces, the number of entries in the buffer is the number of bytes which can range from 1 to 2 <sup>30</sup> . After subtracting one from the number of entries, software must place the fields of the resulting 27-bit value into the <b>Height</b> , <b>Width</b> , and <b>Depth</b> fields as indicated, right-justified in each field. Unused upper bits must be set to zero.			
If <b>Vertical Line Stride</b> is 1, this field indicates the height of the field, not the height of the frame			
The <b>Height</b> of a render target must be the same as the <b>Height</b> of the other render targets and the depth buffer (defined in 3DSTATE_DEPTH_BUFFER), unless <b>Surface Type</b> is SURFTYPE_1D or SURFTYPE_2D with <b>Depth</b> = 0 (non-array) and <b>LOD</b> = 0 (non-mip mapped).			
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.			
If Media Pixel Boundary Mode is not set to NORMAL_MODE, this field must be an even value.			
If Surface Format is PLANAR*, this field must be a multiple of 4			
15:1	<b>Reserved</b>		
4	Access:		RO
	Format:		MBZ
13:0	<b>Width</b>		
	Format:		U14-1
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.			
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	[0,16383]		width of surface - 1 (x/u dimension)
	[0,16383]		width of surface - 1 (x/u dimension)
	[0,16383]		width of surface - 1 (x/u dimension)
	[0,2047]		width of surface - 1 (x/u dimension)
	[0,16383]		width of surface - 1 (x/u dimension)
			[SurfaceType] == 'SURFTYPE_1D'
			[SurfaceType] == 'SURFTYPE_2D'
			[SurfaceType] == 'SURFTYPE_3D'
			[SurfaceType] == 'SURFTYPE_3D'
			[SurfaceType] == 'SURFTYPE_CUBE'

## RENDER\_SURFACE\_STATE

		[0,127]	contains bits [6:0] of the number of entries in the buffer - 1	[SurfaceType] == 'SURFTYPE_BUFFER'
<b>Programming Notes</b>				
<ul style="list-style-type: none"> <li>For surface types other than SURFTYPE_BUFFER. The Width specified by this field must be less than or equal to the surface pitch (specified in bytes via the Surface Pitch field).</li> <li>For cube maps, Width must be set equal to the Height.</li> <li>The <b>Width</b> of a render target must be the same as the <b>Width</b> of the other render target(s) and the depth buffer (defined in 3DSTATE_DEPTH_BUFFER), unless <b>Surface Type</b> is SURFTYPE_1D or SURFTYPE_2D with <b>Depth</b> = 0 (non-array) and <b>LOD</b> = 0 (non-mip mapped).</li> <li>The <b>Width</b> of a render target with YUV surface format must be a multiple of 2.</li> <li>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).</li> </ul>				
If <b>Surface Format</b> 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				
3	31:2 1	<b>Depth</b>		
		Format:	U11-1	
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 <b>Minimum Array Element</b> 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.				
<b>Value</b>	<b>Name</b>	<b>Description</b>	<b>Exists If</b>	
[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	
<b>Programming Notes</b>				
The <b>Depth</b> of a render target must be the same as the <b>Depth</b> of the other render target(s) and of the depth buffer (defined in 3DSTATE_DEPTH_BUFFER).				

<b>RENDER_SURFACE_STATE</b>													
		<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 <b>Minimum Array Element</b>. For example, if <b>Minimum Array Element</b> is set to 1024 on a 2D surface, the range of this field is reduced to [0,1023].</p>											
20	<b>Tile Address Mapping Mode</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U1</td> </tr> </table> <p>This field is used to select between Tile Address Mapping mode and for TileYs and TileYf.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 55%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Tile Address Mapping Mode</td> <td>This field should always be programmed to 0h. Tiling mode is the standard tile layout for 3D.</td> </tr> <tr> <td>1h</td> <td>Tile Address Mapping Mode (for Standard Tiling)</td> <td>This field should NOT be programmed to 1h as the Tiling Address Mapping mode is not supported by SW.</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>Tile Address Mapping Mode must be set to Tile Address Mapping Mode when surface type is SURFTYPE_3D.</p>	Format:	U1	Value	Name	Description	0h	Tile Address Mapping Mode	This field should always be programmed to 0h. Tiling mode is the standard tile layout for 3D.	1h	Tile Address Mapping Mode (for Standard Tiling)	This field should NOT be programmed to 1h as the Tiling Address Mapping mode is not supported by SW.
Format:	U1												
Value	Name	Description											
0h	Tile Address Mapping Mode	This field should always be programmed to 0h. Tiling mode is the standard tile layout for 3D.											
1h	Tile Address Mapping Mode (for Standard Tiling)	This field should NOT be programmed to 1h as the Tiling Address Mapping mode is not supported by SW.											
19	<b>Standard Tiling Mode Extensions</b>	<p>It changes in the MIP Tail Packing. When enabled (programmed to 1h), MIP Tail packing for Volumetric and 1D are changed as defined in the Surface Layout and Tiling section.</p> <p>This bit controls enabling of some Standard Tiling extensions:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable <b>[Default]</b></td> <td>When programmed to 0h, the extensions to support Standard Tiling are disabled. Behavior reverts to Miptail packing.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>When programmed to 1h, the changes to support Standard Tiling Extensions are enabled. See the Surface Layout and Tiling section for details.</td> </tr> </tbody> </table>	Value	Name	Description	0h	Disable <b>[Default]</b>	When programmed to 0h, the extensions to support Standard Tiling are disabled. Behavior reverts to Miptail packing.	1h	Enable	When programmed to 1h, the changes to support Standard Tiling Extensions are enabled. See the Surface Layout and Tiling section for details.		
Value	Name	Description											
0h	Disable <b>[Default]</b>	When programmed to 0h, the extensions to support Standard Tiling are disabled. Behavior reverts to Miptail packing.											
1h	Enable	When programmed to 1h, the changes to support Standard Tiling Extensions are enabled. See the Surface Layout and Tiling section for details.											
18	<b>Reserved</b>	<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												
17:0	<b>Surface Pitch</b>	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U18-1</td> </tr> </table> <p style="text-align: center;"><b>Description</b></p> <p>Surface Pitch Range:</p> <ul style="list-style-type: none"> <li>• For surfaces of type SURFTYPE_BUFFER: [0,262143] -&gt; [1B, 256KB]</li> <li>• For other linear surfaces: [0, 262143] -&gt; [1B, 256KB]</li> <li>• For X-tiled surface: [511, 262143] -&gt; [512B, 256KB] = [1 tile, 512 tiles]</li> </ul>	Format:	U18-1									
Format:	U18-1												

## RENDER\_SURFACE\_STATE

- For Y-tiled surfaces: [127, 262143]->[128B, 256KB] = [1 tile, 2048 tiles]
- 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<sup>Cu</sup>) tiles]

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

For surfaces of type SURFTYPE\_BUFFER, 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 must match the number of bytes implied by Surface Format..
- 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.

- 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<sup>Cu</sup> 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
	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

RENDER_SURFACE_STATE						
			16bpp	16k	8k	16k + 511
			32bpp	16k	4k	16k + 511
			64bpp	16k	2k	16k + 1023
			128bpp	16k	1k	16k + 1023
4	31	<b>Decompress In L3</b>				
		<b>Value</b>	<b>Name</b>	<b>Description</b>		
		0h	Disable <b>[Default]</b>	When this field is set to 0h, the associated compressible surface, when accessed by sampler and data-port, can be compressed in L3. If the surface is not compressible, this bit field is ignored.		
		1h	Enable	When this field is set to 1h, the associated compressible surface, when accessed by sampler and data-port, will be uncompressed in L3. If the surface is not compressible, this bit field is ignored.		
30:29	9	<b>Render Target And Sample Unorm Rotation</b>				
		<b>Description</b>				
		<b>For Render Target Surfaces:</b> 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.				
		<b>For Other Surfaces:</b> This field is ignored.				
		<b>Value</b>	<b>Name</b>	<b>Description</b>		
		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		
		<b>Programming Notes</b>				
		<b>Programming Notes for Render Target Surfaces only</b>				
		<ul style="list-style-type: none"> <li>• 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.</li> <li>• <b>Width</b> and <b>Height</b> fields apply to the dimensions of the surface before rotation.</li> <li>• For 90 and 270 degree rotated surfaces, the <b>Height</b> (rather than the <b>Width</b>) must be less than or equal to the <b>Surface Pitch</b> (specified in bytes).</li> <li>• For 90 and 270 degree rotated surfaces, the actual <b>Height</b> and <b>Width</b> of the surface in pixels (not the field value which is decremented) must both be even.</li> </ul>				
		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, R10G10B10A2_UNORM,				

## RENDER\_SURFACE\_STATE

		R16G16B16A16_FLOAT, R16G16B16X16_FLOAT	
28:1 8	<b>Minimum Array Element</b>		
	Format:	U11	
	<p>Range 1D/2D/cube surfaces: [0,2047] 3D surfaces: [0,2047]</p> <p><b>For Sampling Engine, Render Target, and Typed 1D, 2D and 3D Surfaces:</b> This field indicates the minimum array element that can be accessed as part of this surface. This field is added to the delivered array index before it is used to address the surface. For sampling 3D surfaces, this field is an offset in "R" slices into the LOD.</p> <p><b>For Sampling Engine Cube Surfaces:</b> This field indicates the minimum array element in units of "cube faces" in the underlying 2D surface array that can be accessed as part of this surface (the cube array index is multiplied by 6 to compute this value, although Minimum Array Element is not restricted to only multiples of 6). This field is added to the delivered (array index)*6 before it is used to address the surface.</p> <p><b>For all other surface types:</b> This field should be zero.</p>		
17:7	<b>Render Target View Extent</b>		
	Format:	U11-1	
	<p>Range [0,2047] to indicate extent of [1,2048]</p> <p><b>For Render Target and Typed Dataport 3D Surfaces:</b> This field indicates the extent of the accessible 'R' coordinates minus 1 on the LOD currently being rendered to.</p> <p><b>For Render Target and Typed Dataport 1D and 2D Surfaces:</b> This field must be set to the same value as the Depth field.</p> <p><b>For Other Surfaces:</b> This field is ignored.</p>		
6	<b>Multisampled Surface Storage Format</b>		
	This field indicates the storage format of the multisampled surface.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	MSS	Multisampled surface was/is rendered as a render target
	1h	DEPTH_STENCIL	Multisampled surface was rendered as a depth or stencil buffer
	<b>Programming Notes</b>		
	<ul style="list-style-type: none"> <li>All multisampled render target surfaces must have this field set to MSFMT_MSS</li> <li>IF this field is MSFMT_DEPTH_STENCIL, the only sampling engine messages allowed are "ld2dms", "resinfo", and "sampleinfo".</li> <li>This field is ignored if <b>Number of Multisamples</b> is MULTISAMPLECOUNT_1</li> </ul>		
5:3	<b>Number of Multisamples</b>		
	This field indicates the number of multisamples on the surface.		
	<b>Value</b>	<b>Name</b>	
	0h	MULTISAMPLECOUNT_1	

<b>RENDER_SURFACE_STATE</b>														
		<table border="1"> <tr><td>1h</td><td>MULTISAMPLECOUNT_2</td></tr> <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> <p style="text-align: center;"><b>Programming Notes</b></p> <p>If this field is any value other than MULTISAMPLECOUNT_1, the <b>Surface Type</b> 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>	1h	MULTISAMPLECOUNT_2	2h	MULTISAMPLECOUNT_4	3h	MULTISAMPLECOUNT_8	4h	MULTISAMPLECOUNT_16	5h-7h	Reserved		
1h	MULTISAMPLECOUNT_2													
2h	MULTISAMPLECOUNT_4													
3h	MULTISAMPLECOUNT_8													
4h	MULTISAMPLECOUNT_16													
5h-7h	Reserved													
	2:0	<p><b>Multisample Position Palette Index</b></p> <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>	Value	Name	[0,7]									
Value	Name													
[0,7]														
5	31:2 5	<p><b>X Offset</b></p> <table border="1"> <tr> <td>Format:</td> <td>U9[8:2]</td> </tr> </table> <p>This field specifies the horizontal offset in pixels from the <b>Surface Base Address</b> 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"> <tr> <td colspan="2">Format:</td> </tr> <tr> <td colspan="2">PixelOffset[8:2]</td> </tr> </table> <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>[0,127]</td> <td></td> <td>This value is multiplied by 4 to determine X Offset in pixels.</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>• For linear surfaces, this field must be zero.</li> <li>• 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.</li> <li>• For surfaces accessed with the <b>Data Port Transpose Read message</b>, the pixel size is assumed to be 32 bits in width.</li> <li>• For <b>Surface Format</b> with other than 8, 16, 32, 64, or 128 bits per pixel, this field must be zero.</li> <li>• If <b>Render Target Rotation</b> is set to other than RTROTATE_0DEG, this field must be zero.</li> </ul>	Format:	U9[8:2]	Format:		PixelOffset[8:2]		Value	Name	Description	[0,127]		This value is multiplied by 4 to determine X Offset in pixels.
Format:	U9[8:2]													
Format:														
PixelOffset[8:2]														
Value	Name	Description												
[0,127]		This value is multiplied by 4 to determine X Offset in pixels.												



## RENDER\_SURFACE\_STATE

- 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.

If Surface Format is PLANAR\_420\_16 or PLANAR\_420\_8 this field must be zero

- If **Tiled Resource Mode** is not TRMODE\_NONE, this field must be zero.

24 **Reserved**

Access:	RO
Format:	MBZ

23:2 **Y Offset**

1

Format:	U5[4:2]
---------	---------

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.)

Format:

RowOffset [4:2]

Value	Name	Description
[0,7]		This value is multiplied by 4 to determine the actual Y Offset in Rows.

### Programming Notes

- 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.

## RENDER\_SURFACE\_STATE

		If Surface Format is PLANAR_420_16 or PLANAR_420_8 this field must be zero	
		<ul style="list-style-type: none"> <li>If <b>Tiled Resource Mode</b> is not TRMODE_NONE, this field must be zero.</li> </ul>	
		This field must be zero if Surface Format is Planar and the U and V planes are half-pitch (e.g. YV12 format).	
		If VERTICAL STRIDE field is 1h, then the Y Offset must be a multiple of 8. Therefore, the lsb of this field must be 0.	
20	<b>EWA Disable For Cube</b>	Format: <span style="float: right;">Disable</span>	
		Specifies if EWA mode for LOD quality improvement needs to be disabled for cube maps.	
		<b>Value</b>	<b>Name</b>
		0h	Enable <b>[Default]</b>
		1h	Disable
		<b>Description</b>	
		EWA is enabled for cube maps	
		EWA is disabled for cube maps	
		<b>Programming Notes</b>	
		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.	
19:1 8	<b>Tiled Resource Mode For Sampling Engine, Render Target, and Typed/Untyped Surfaces:</b>	This field specifies the tiled resource mode. <b>For other surfaces:</b> This field is ignored.	
		<b>Value</b>	<b>Name</b>
		0h	NONE
		1h	4KB
		2h	64KB
		1h	TILEYF
		2h	TILEYS
		3h	Reserved
		<b>Description</b>	
		No tiled resource	
		4KB tiled resources	
		64KB tiled resources	
		4KB tiled resources	
		64KB tiled resources	
		<b>Exists If</b>	
		[SurfaceType] == 'SURFTYPE_1D'	
		[SurfaceType] == 'SURFTYPE_1D'	
		[SurfaceType] != 'SURFTYPE_1D'	
		[SurfaceType] != 'SURFTYPE_1D'	
		<b>Programming Notes</b>	
		If <b>Tile Mode</b> 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 <b>Surface Format</b> 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 <b>Separate UV Plane Enable</b> is disabled) or 8 bits per element on the U and V planes (if <b>Separate UV Plane Enable</b> is enabled).	
		If this field is set to TRMODE_NONE, the surface cannot contain any null pages unless <b>Surface Type</b> is BUFFER. A BUFFER surface with null pages must have <b>Surface Base Address</b> and <b>Surface</b>	

## RENDER\_SURFACE\_STATE

		<p><b>Pitch</b> set to an integer multiple of the element size, and <b>Surface Format</b> must be one with 8, 16, 32, 64, or 128 bits per element.</p> <p>If <b>Surface Format</b> is PLANAR, the surface cannot contain any null pages.</p> <p>If RENDER_SURFACE_STATE.Surface Type = 3D, and RENDER_SURFACES_STATE.Auxiliary Surface Mode != AUX_NONE , this field must be set to TILEYS</p>	
17:1	<b>Reserved</b>		
5	Access:	RO	
	Format:	MBZ	
14	<b>Coherency Type</b>	Specifies the type of coherency maintained for this surface.	
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	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).
	<b>Programming Notes</b>		
	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).		
13:1	<b>Reserved</b>		
2	Access:	RO	
	Format:	MBZ	
11:8	<b>Mip Tail Start LOD</b>		
	Format:	U4	
	<b>Description</b>		
	<p><b>For Sampling Engine, Render Target, and Typed Surfaces:</b> This field indicates which LOD is the first one in the MIP tail if <b>Tiled Resource Mode</b> 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. <b>For other surfaces:</b> This field is ignored.</p>		
	<b>Programming Notes</b>		
	If <b>Tiled Resource Mode</b> 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 <b>Tiled Resource Mode</b> 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-		

## RENDER\_SURFACE\_STATE

mapped surfaces.

The following table indicates the *maximum* 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/ CUBE	TlleYS	1	128x256	128x128	64x128	64x64	32x64
		2	128x128	128x64	64x64	64x32	32x32
		4	64x128	64x64	32x64	32x32	16x32
		8	64x64	64x32	32x32	32x16	16x16
		16	32x64	32x32	16x32	16x16	8x16
	TlleYF	1x	32x64	32x32	16x32	16x16	8x16
		2x	16x32	16x16	8x16	8x8	4x8
		4x	16x16	16x8	8x8	8x4	4x4
		8x	16x16	16x8	8x8	8x4	4x4
		16x	8x4	8x4	4x4	2x4	1x4
3D	TlleYS	1	32x32x32	16x32x32	16x32x16	16x16x16	8x16x16
	TileYF	1	16x8x16	8x8x16	8x8x8	8x4x8	4x4x8

**7:4 Surface Min LOD**

Format:	U4
---------	----

**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 (*sample\_l*, *ld*, or *resinfo* message types) before it is used to address the surface.

**For Other Surfaces:** This field is ignored.

**3:0 MIP Count / LOD**

Format:	<b>Sampling Engine and Typed Surfaces:</b> U4 in (LOD units - 1) <b>Render Target Surfaces:</b> U4 in LOD units
---------	---

Range	<b>Sampling Engine and Typed Surfaces:</b> [0,14] representing [1,15] MIP levels <b>Render Target Surfaces:</b> [0,14] representing LOD <b>Other Surfaces:</b> [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 *sample\** 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 *ld\** 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.

<b>RENDER_SURFACE_STATE</b>																	
	<p><b>For Other Surfaces:</b> This field is reserved : MBZ</p> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> <p>The <b>LOD</b> of a render target must be the same as the <b>LOD</b> of the other render target(s) and of the depth buffer (defined in 3DSTATE_DEPTH_BUFFER).</p> <p>For render targets with YUV surface formats, the <b>LOD</b> must be zero.</p> <p>For sampling engine surfaces with PLANAR* surface format, <b>MIP Count</b> must be zero.</p> </td> </tr> </tbody> </table>	Programming Notes		<p>The <b>LOD</b> of a render target must be the same as the <b>LOD</b> of the other render target(s) and of the depth buffer (defined in 3DSTATE_DEPTH_BUFFER).</p> <p>For render targets with YUV surface formats, the <b>LOD</b> must be zero.</p> <p>For sampling engine surfaces with PLANAR* surface format, <b>MIP Count</b> must be zero.</p>													
Programming Notes																	
<p>The <b>LOD</b> of a render target must be the same as the <b>LOD</b> of the other render target(s) and of the depth buffer (defined in 3DSTATE_DEPTH_BUFFER).</p> <p>For render targets with YUV surface formats, the <b>LOD</b> must be zero.</p> <p>For sampling engine surfaces with PLANAR* surface format, <b>MIP Count</b> must be zero.</p>																	
6	<p>31 <b>Separate UV Plane Enable</b></p> <table border="1" style="width: 100%;"> <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> <table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2"> <p>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> </td> </tr> </tbody> </table>	Exists If:	([Surface Format] == 'PLANAR')	Format:	Enable	Programming Notes		<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																
Programming Notes																	
<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>																	
	<p>30 <b>Half Pitch for Chroma</b></p> <table border="1" style="width: 100%;"> <tr> <td>Exists If:</td> <td>([Surface Format] == 'PLANAR')</td> </tr> </table> <p>This bit enables support for half-pitch chroma planes for Planar YUV surfaces. It is ignored for Non-Planar surfaces. For planar surfaces it allows the chroma planes to be one-half the width of a the Y (Luma) plane.</p> <p>For example, should be set to 0h for NV12 surfaces.</p> <p>Must be set to 1h for YV12 surfaces.</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>Disable <b>[Default]</b></td> <td>Setting this bit to 0h (default) causes Chroma planes to be treated as full width (same as Y plane).</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Setting this bit to 1h causes Chroma planes (U and V) to be treated as half the width of the Luma (Y) plane.</td> </tr> </tbody> </table>	Exists If:	([Surface Format] == 'PLANAR')	Value	Name	Description	0h	Disable <b>[Default]</b>	Setting this bit to 0h (default) causes Chroma planes to be treated as full width (same as Y plane).	1h	Enable	Setting this bit to 1h causes Chroma planes (U and V) to be treated as half the width of the Luma (Y) plane.					
Exists If:	([Surface Format] == 'PLANAR')																
Value	Name	Description															
0h	Disable <b>[Default]</b>	Setting this bit to 0h (default) causes Chroma planes to be treated as full width (same as Y plane).															
1h	Enable	Setting this bit to 1h causes Chroma planes (U and V) to be treated as half the width of the Luma (Y) plane.															
30:1	<p>6 <b>Auxiliary Surface QPitch</b></p> <table border="1" style="width: 100%;"> <tr> <td>Exists If:</td> <td>([Surface Format] != 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>U17[16:2]</td> </tr> </table> <p>This field specifies the distance in rows between array slices on the auxiliary surface.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0h,7FFFh]</td> <td></td> <td>1 is added to the value and it is multiplied by 4 to determine the actual QPitch in rows</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"> <p>This field must be set to an integer multiple of the <b>Surface Vertical Alignment</b></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> </td> </tr> <tr> <td colspan="2"> <p>Auxiliary surface parameters in the Surface State is only meant for Multisampling Control Surface.</p> </td> </tr> </tbody> </table>	Exists If:	([Surface Format] != 'PLANAR')	Format:	U17[16:2]	Value	Name	Description	[0h,7FFFh]		1 is added to the value and it is multiplied by 4 to determine the actual QPitch in rows	Programming Notes		<p>This field must be set to an integer multiple of the <b>Surface Vertical Alignment</b></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>Auxiliary surface parameters in the Surface State is only meant for Multisampling Control Surface.</p>	
Exists If:	([Surface Format] != 'PLANAR')																
Format:	U17[16:2]																
Value	Name	Description															
[0h,7FFFh]		1 is added to the value and it is multiplied by 4 to determine the actual QPitch in rows															
Programming Notes																	
<p>This field must be set to an integer multiple of the <b>Surface Vertical Alignment</b></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>Auxiliary surface parameters in the Surface State is only meant for Multisampling Control Surface.</p>																	

## RENDER\_SURFACE\_STATE

	<p>This field should be programmed when Multisampling is enabled. In all other cases HW will ignore this value.</p>														
29:1	<p><b>X Offset for U or UV Plane</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">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 <b>Surface Base Address</b> to the start (origin) of the U plane or interleaved UV plane, depending on the setting of <b>Separate UV Plane Enable</b>.</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>This field must be a multiple of 4 (bits 1:0 MBZ).</td> </tr> <tr> <td>If <b>Tiled Resource Mode</b> is enabled, this field must be a multiple of the tile width in pixels.</td> </tr> <tr> <td><b>Auxiliary Surface Mode</b> is forced to AUX_NONE.</td> </tr> </table>		Exists If:	([Surface Format] == 'PLANAR')	Format:	U14	Programming Notes	This field must be a multiple of 4 (bits 1:0 MBZ).	If <b>Tiled Resource Mode</b> is enabled, this field must be a multiple of the tile width in pixels.	<b>Auxiliary Surface Mode</b> is forced to AUX_NONE.					
Exists If:	([Surface Format] == 'PLANAR')														
Format:	U14														
Programming Notes															
This field must be a multiple of 4 (bits 1:0 MBZ).															
If <b>Tiled Resource Mode</b> is enabled, this field must be a multiple of the tile width in pixels.															
<b>Auxiliary Surface Mode</b> is forced to AUX_NONE.															
15	<p><b>YUV Interpolation Enable</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; background-color: #e1eef6;">Description</th> </tr> <tr> <td>This bit controls whether a Non-Planar YUV4:2:2 and Planar YUV4:2:0 surface use interpolated or replicated U and V channels for input to the Sampler filter. Programming to 1h causes interpolation of U and V channels. In this case the chrominance for odd pixels is computed by an interpolation between adjacent even pixels. Programming to 0h causes the chrominance to be copied from the pixel to the left.</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; background-color: #e1eef6;">Value</th> <th style="text-align: center; background-color: #e1eef6;">Name</th> <th style="text-align: center; background-color: #e1eef6;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td>Disable <b>[Default]</b></td> <td>Programming to 0h causes the sampler to replicate U and V channels. This will lead to lower quality in certain cases where the YUV surface is being filtered (e.g. linear).</td> </tr> <tr> <td style="text-align: center;">1h</td> <td>Enable</td> <td>Programming to 1h causes the sampler to interpolate the U and V channels between the horizontally neighboring pixels. This will improve image quality if the surface is being filtered.</td> </tr> </tbody> </table>		Format:	Enable	Description	This bit controls whether a Non-Planar YUV4:2:2 and Planar YUV4:2:0 surface use interpolated or replicated U and V channels for input to the Sampler filter. Programming to 1h causes interpolation of U and V channels. In this case the chrominance for odd pixels is computed by an interpolation between adjacent even pixels. Programming to 0h causes the chrominance to be copied from the pixel to the left.	Value	Name	Description	0h	Disable <b>[Default]</b>	Programming to 0h causes the sampler to replicate U and V channels. This will lead to lower quality in certain cases where the YUV surface is being filtered (e.g. linear).	1h	Enable	Programming to 1h causes the sampler to interpolate the U and V channels between the horizontally neighboring pixels. This will improve image quality if the surface is being filtered.
Format:	Enable														
Description															
This bit controls whether a Non-Planar YUV4:2:2 and Planar YUV4:2:0 surface use interpolated or replicated U and V channels for input to the Sampler filter. Programming to 1h causes interpolation of U and V channels. In this case the chrominance for odd pixels is computed by an interpolation between adjacent even pixels. Programming to 0h causes the chrominance to be copied from the pixel to the left.															
Value	Name	Description													
0h	Disable <b>[Default]</b>	Programming to 0h causes the sampler to replicate U and V channels. This will lead to lower quality in certain cases where the YUV surface is being filtered (e.g. linear).													
1h	Enable	Programming to 1h causes the sampler to interpolate the U and V channels between the horizontally neighboring pixels. This will improve image quality if the surface is being filtered.													
14	<p><b>Reserved</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ									
Access:	RO														
Format:	MBZ														
13:0	<p><b>Y Offset for U or UV Plane</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">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 <b>Surface Base Address</b> to the start (origin) of the U plane or interleaved UV plane, depending on the setting of <b>Separate UV Plane Enable</b>.</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>For surfaces where <b>Surface Format</b> = PLANAR* and <b>Separate UV Plane</b> is Enabled, the Y Offset must be programmed in multiples of <b>half-rows</b>. For example, for a surface where Y is physically</td> </tr> </table>		Exists If:	([Surface Format] == 'PLANAR')	Format:	U14	Programming Notes	For surfaces where <b>Surface Format</b> = PLANAR* and <b>Separate UV Plane</b> is Enabled, the Y Offset must be programmed in multiples of <b>half-rows</b> . For example, for a surface where Y is physically							
Exists If:	([Surface Format] == 'PLANAR')														
Format:	U14														
Programming Notes															
For surfaces where <b>Surface Format</b> = PLANAR* and <b>Separate UV Plane</b> is Enabled, the Y Offset must be programmed in multiples of <b>half-rows</b> . For example, for a surface where Y is physically															

## RENDER\_SURFACE\_STATE

	<p>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.</p> <p>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.</p> <p>If <b>Tiled Resource Mode</b> is enabled, this field must be a multiple of the tile height in rows.</p> <p><b>Auxiliary Surface Mode</b> is forced to AUX_NONE.</p>																									
	<p><b>Workaround</b></p> <p>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.</p>																									
11:3	<p><b>Auxiliary Surface Pitch</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Exists If:</td> <td>([Surface Format] != 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>U9-1</td> </tr> </table> <p>This field specifies the Auxiliary surface pitch in (#Tiles - 1).</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, 511]</td> <td></td> <td>-&gt; [1 tile, 512 tiles]</td> </tr> </tbody> </table>	Exists If:	([Surface Format] != 'PLANAR')	Format:	U9-1	Value	Name	Description	[0, 511]		-> [1 tile, 512 tiles]															
Exists If:	([Surface Format] != 'PLANAR')																									
Format:	U9-1																									
Value	Name	Description																								
[0, 511]		-> [1 tile, 512 tiles]																								
2:0	<p><b>Auxiliary Surface Mode</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Exists If:</td> <td>([Surface Format] != 'PLANAR')</td> </tr> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 70%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>AUX_NONE</td> <td>No Auxiliary surface is used</td> </tr> <tr> <td>1h</td> <td>AUX_CCS_D</td> <td>To be programmed only for Procedural Texture (PT). This mode is not valid for Render Targets.</td> </tr> <tr> <td>2h</td> <td>AUX_APPEND</td> <td>The Auxiliary surface is an append buffer</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> <tr> <td>4h</td> <td>AUX_MCS_LCE</td> <td> <p>[]</p> <p>Enables lossless compression on the top of MSAA compression for RTs with Number of Multisamples not equal to MULTISAMPLECOUNT_1. CCS is tiling format is linear. MCS tiling format is always tiledY.</p> </td> </tr> <tr> <td>5h</td> <td>AUX_CCS_E</td> <td> <p>If Number of multisamples = 1, programming this value means lossless compression is enabled for that surface. Auxiliary surface is a CCS with linear tiling.</p> <p>AUX_CCS_E cannot be programmed for Procedural Texture surfaces</p> </td> </tr> </tbody> </table>	Exists If:	([Surface Format] != 'PLANAR')	Format:	U3	Value	Name	Description	0h	AUX_NONE	No Auxiliary surface is used	1h	AUX_CCS_D	To be programmed only for Procedural Texture (PT). This mode is not valid for Render Targets.	2h	AUX_APPEND	The Auxiliary surface is an append buffer	3h	Reserved		4h	AUX_MCS_LCE	<p>[]</p> <p>Enables lossless compression on the top of MSAA compression for RTs with Number of Multisamples not equal to MULTISAMPLECOUNT_1. CCS is tiling format is linear. MCS tiling format is always tiledY.</p>	5h	AUX_CCS_E	<p>If Number of multisamples = 1, programming this value means lossless compression is enabled for that surface. Auxiliary surface is a CCS with linear tiling.</p> <p>AUX_CCS_E cannot be programmed for Procedural Texture surfaces</p>
Exists If:	([Surface Format] != 'PLANAR')																									
Format:	U3																									
Value	Name	Description																								
0h	AUX_NONE	No Auxiliary surface is used																								
1h	AUX_CCS_D	To be programmed only for Procedural Texture (PT). This mode is not valid for Render Targets.																								
2h	AUX_APPEND	The Auxiliary surface is an append buffer																								
3h	Reserved																									
4h	AUX_MCS_LCE	<p>[]</p> <p>Enables lossless compression on the top of MSAA compression for RTs with Number of Multisamples not equal to MULTISAMPLECOUNT_1. CCS is tiling format is linear. MCS tiling format is always tiledY.</p>																								
5h	AUX_CCS_E	<p>If Number of multisamples = 1, programming this value means lossless compression is enabled for that surface. Auxiliary surface is a CCS with linear tiling.</p> <p>AUX_CCS_E cannot be programmed for Procedural Texture surfaces</p>																								

## RENDER\_SURFACE\_STATE

		If Number of multisamples > 1, programming this value means MSAA compression is enabled for that surface. Auxillary surafce is MSC with tile y.
6h	Reserved	
7h	Reserved	

### Programming Notes

The CCS surface shares **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. **X & Y Offset** are set to zero for the purpose of accessing the Auxiliary surface. CCS Height and Width are scaled.

CCS is always linear.

MCS is always tile-y.

CCS surfaces are linear 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 **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 **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, value of AUX\_CCS\_E is only allowed value if **Surface Format** is supported for Render Target Compression.

If **Number of Multisamples** is other than MULTISAMPLECOUNT\_1, value of AUX\_MCS\_LCE is only allowed if **Surface Format** is supported for Render Target Compression.

If **Number of Multisamples** is MULTISAMPLECOUNT\_1, and if **Tiled Resource Mode** is NOT TRMODE\_NONE, then, if CCS tile is NULL, Render Target Tiles represented by that CCS tile are assumed to be NULL by HW.

When Number of Multisamples > 1, and when MSAA compression is disabled, there is no way to enable lossless compression.

SW can enable lossless compression for MSRTs, by setting this the field to AUX\_MCS\_LCE for Render Targets with **Number of Multisamples** *not* equal to MULTISAMPLECOUNT\_1.

This value can not be progammed when **Number of Multisamples** equal to MULTISAMPLECOUNT\_1.

Programming AUX\_CCS\_D is not allowed for all surfaces EXCEPT Procedural Texture (PT).

For an 8 bpp surface with NUM\_MULTISAMPLES = 1, Surface Width not multiple of 64 pixels and more than 1 mip level in the view, Fast Clear is not supported when AUX\_CCS\_E is set in this field.

For Surface Type = SURFACETYPE\_3D, Surface Format = 64bpp or 128bpp and Tiling Format = TiledYs, AUX\_MODE must not be programmed to enable lossless compression or attaching CCS i.e. AUX\_MODE must be programmed to AUX\_NONE. except when Procedural Texture Enable bit is set. For Procedural Textures, Surface format TYS, 3D, 64/128 bpp is supported

ValidValue	Valu	Description	Msaa	CCS	Color	MCS	Msaa
------------	------	-------------	------	-----	-------	-----	------



## RENDER\_SURFACE\_STATE

	e		Mode	Aux Present	Compression Enabled	Aux Present	Compression Enabled
AUX_NONE	0	No Auxiliary surface is used					
AUX_CCS_D	1	To be programmed only for Procedural Texture (PT). This mode is not valid for Render Targets.	NA	True	False	False	False
AUX_APPEND	2	The Auxiliary surface is an append buffer	NA	-	-	-	-
RESERVED	3	-	-	-	-	-	-
AUX_MCS_LC_E	4	Enables lossless compression on the top of MSAA compression for RTs with <b>Number of Multisamples</b> <i>not</i> equal to MULTISAMPLECOUNT_1.	nX	True	True	True	True
AUX_CCS_E	5	If Number of multisamples = 1, programming this value means lossless compression is enabled for that surface. Auxiliary surface is a CCS with linear tiling.	1X	True	True	False	False
AUX_CCS_E	5	If Number of multisamples > 1, programming this value means MSAA compression is enabled for that surface. Auxiliary surface is MSC with tile y.	nX	False	False	True	True
*nX = 2x, 4x, 8x, 16x							
7	31	<b>Memory Compression Mode</b>					
Default Value:				0 Horizontal			

<b>RENDER_SURFACE_STATE</b>							
	<table border="1"> <tr> <th colspan="2" style="text-align: center;">Description</th> </tr> <tr> <td colspan="2">MBZ: Only Horizontal mode is supported.</td> </tr> </table>	Description		MBZ: Only Horizontal mode is supported.			
Description							
MBZ: Only Horizontal mode is supported.							
30	<p><b>Memory Compression Enable</b></p> <table border="1"> <tr> <td>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"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Please refer to vol1a Memory Data Formats chapter section Media Memory Compression for more details, including format restrictions.</td> </tr> </table>	Format:	Enable	Programming Notes		Please refer to vol1a Memory Data Formats chapter section Media Memory Compression for more details, including format restrictions.	
Format:	Enable						
Programming Notes							
Please refer to vol1a Memory Data Formats chapter section Media Memory Compression for more details, including format restrictions.							
29:2 8	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
Access:	RO						
Format:	MBZ						
27:2 5	<p><b>Shader Channel Select Red</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>Shader Channel Select</b></td> </tr> </table> <p>Specifies which surface channel is read or written in the Red shader channel.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2"> <p>For reads, the Shader Channel Select defines which surface channel is filled into the shader channel. If the Shader Channel Select is SCS_ZERO or SCS_ONE, then the shader channel takes on that fixed value, regardless of the surface channel's presence or value. Otherwise, if the Shader Channel Select is SCS_RED, SCS_GREEN, SCS_BLUE or SCS_ALPHA, then that surface channel is read into this shader channel. If the same Surface Channel Select appears for multiple shader channels, then that surface channel is replicated in each of those shader channels. If a Shader Channel Select is specified that is not present in the surface format, the value filled into the shader channel is undefined. Programs should always use SCS_ZERO or SCS_ONE for missing surface channels.</p> <p>For writes, the Shader Channel Select defines which surface channel is written from the shader channel. If the Shader Channel Select is not present in the surface format or is SCS_ZERO or SCS_ONE, then the shader channel is not written to the surface. Otherwise, if the Shader Channel Select is SCS_RED, SCS_GREEN, SCS_BLUE or SCS_ALPHA, the shader channel is written to that surface channel. 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 the surface channel, and subsequent shader channels with that Shader Channel Select are not written to the surface. If any surface channels are present but not specified by the Shader Channel Select, those surface channels are undefined (and might be written to zero). Programs should always specify the Shader Channel Select to cover all present surface channels.</p> <p>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 sample_unorm* or sample_8x8 messages.</p> <p>The Shader Channel Select fields do not affect the following sampling engine message types: resinfo, sampleinfo, LOD, and Id_mcs. These messages behave as if each Shader Channel Select is set to the same color surface channel.</p> </td> </tr> </table>	Format:	<b>Shader Channel Select</b>	Programming Notes		<p>For reads, the Shader Channel Select defines which surface channel is filled into the shader channel. If the Shader Channel Select is SCS_ZERO or SCS_ONE, then the shader channel takes on that fixed value, regardless of the surface channel's presence or value. Otherwise, if the Shader Channel Select is SCS_RED, SCS_GREEN, SCS_BLUE or SCS_ALPHA, then that surface channel is read into this shader channel. If the same Surface Channel Select appears for multiple shader channels, then that surface channel is replicated in each of those shader channels. If a Shader Channel Select is specified that is not present in the surface format, the value filled into the shader channel is undefined. Programs should always use SCS_ZERO or SCS_ONE for missing surface channels.</p> <p>For writes, the Shader Channel Select defines which surface channel is written from the shader channel. If the Shader Channel Select is not present in the surface format or is SCS_ZERO or SCS_ONE, then the shader channel is not written to the surface. Otherwise, if the Shader Channel Select is SCS_RED, SCS_GREEN, SCS_BLUE or SCS_ALPHA, the shader channel is written to that surface channel. 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 the surface channel, and subsequent shader channels with that Shader Channel Select are not written to the surface. If any surface channels are present but not specified by the Shader Channel Select, those surface channels are undefined (and might be written to zero). Programs should always specify the Shader Channel Select to cover all present surface channels.</p> <p>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 sample_unorm* or sample_8x8 messages.</p> <p>The Shader Channel Select fields do not affect the following sampling engine message types: resinfo, sampleinfo, LOD, and Id_mcs. These messages behave as if each Shader Channel Select is set to the same color surface channel.</p>	
Format:	<b>Shader Channel Select</b>						
Programming Notes							
<p>For reads, the Shader Channel Select defines which surface channel is filled into the shader channel. If the Shader Channel Select is SCS_ZERO or SCS_ONE, then the shader channel takes on that fixed value, regardless of the surface channel's presence or value. Otherwise, if the Shader Channel Select is SCS_RED, SCS_GREEN, SCS_BLUE or SCS_ALPHA, then that surface channel is read into this shader channel. If the same Surface Channel Select appears for multiple shader channels, then that surface channel is replicated in each of those shader channels. If a Shader Channel Select is specified that is not present in the surface format, the value filled into the shader channel is undefined. Programs should always use SCS_ZERO or SCS_ONE for missing surface channels.</p> <p>For writes, the Shader Channel Select defines which surface channel is written from the shader channel. If the Shader Channel Select is not present in the surface format or is SCS_ZERO or SCS_ONE, then the shader channel is not written to the surface. Otherwise, if the Shader Channel Select is SCS_RED, SCS_GREEN, SCS_BLUE or SCS_ALPHA, the shader channel is written to that surface channel. 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 the surface channel, and subsequent shader channels with that Shader Channel Select are not written to the surface. If any surface channels are present but not specified by the Shader Channel Select, those surface channels are undefined (and might be written to zero). Programs should always specify the Shader Channel Select to cover all present surface channels.</p> <p>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 sample_unorm* or sample_8x8 messages.</p> <p>The Shader Channel Select fields do not affect the following sampling engine message types: resinfo, sampleinfo, LOD, and Id_mcs. These messages behave as if each Shader Channel Select is set to the same color surface channel.</p>							

## RENDER\_SURFACE\_STATE

		<p>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.</p> <p>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.</p>	
		<b>Restriction</b>	
		<p>For all Render Target and HDC messages, the Surface Channel Select is restricted to a subset of combinations that ensures, when reading the surface and then writing that value under the same Surface Channel Select, the identical value is put in memory. Any combination of Surface Channel Selects and channel surface format that do not guarantee this isomorphic property are not supported and may produce undefined results.</p> <p>The isomorphism property is guaranteed by these restrictions:</p> <ul style="list-style-type: none"> <li>When the channel (R, G, B, A) is present in the surface format, then that channel (SCS_RED, SCS_GREEN, SCS_BLUE, SCS_ALPHA) is present exactly once in the 4 Surface Channel Selects (Red, Green, Blue, Alpha).</li> <li>For channels not present in the surface format, the corresponding Surface Channel Select is either SCS_ZERO or SCS_ONE.</li> <li>The Surface Channel Select for present channels either preserves or swaps the order of the surface channels. This guarantees <math>\text{write\_swizzled}(\text{read\_swizzled}(x)) = x</math>.</li> </ul> <p>Render Target messages do not support swapping of colors with alpha. The Red, Green, or Blue Shader Channel Selects do not support SCS_ALPHA. The Shader Channel Select Alpha does not support SCS_RED, SCS_GREEN, or SCS_BLUE.</p>	
24:2	<b>Shader Channel Select Green</b>		
2	Format:	<b>Shader Channel Select</b>	
	See <b>Shader Channel Select Red</b> for details.		
21:1	<b>Shader Channel Select Blue</b>		
9	Format:	<b>Shader Channel Select</b>	
	See <b>Shader Channel Select Red</b> for details.		
18:1	<b>Shader Channel Select Alpha</b>		
6	Format:	<b>Shader Channel Select</b>	
	See <b>Shader Channel Select Red</b> for details.		
	<b>Programming Notes</b>		
	For Render Target, this field MUST be programmed to value = SCS_ALPHA.		
15:1	<b>Reserved</b>		
4	Access:	RO	
	Format:	MBZ	

<b>RENDER_SURFACE_STATE</b>										
	13:1 2	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
	Access:	RO								
Format:	MBZ									
	11:0	<b>Resource Min LOD</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U4.8</td> </tr> </table> <p><b>For Sampling Engine Surfaces:</b>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><b>For Other Surfaces:</b>This field is ignored.</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>[0,14]</td> <td></td> </tr> </tbody> </table> <table border="1" style="width: 100%; text-align: center;"> <tr> <td><b>Programming Notes</b></td> </tr> <tr> <td>This field must be zero if the <b>ChromaKey Enable</b> is enabled in the associated sampler.</td> </tr> </table>	Format:	U4.8	Value	Name	[0,14]		<b>Programming Notes</b>	This field must be zero if the <b>ChromaKey Enable</b> is enabled in the associated sampler.
Format:	U4.8									
Value	Name									
[0,14]										
<b>Programming Notes</b>										
This field must be zero if the <b>ChromaKey Enable</b> is enabled in the associated sampler.										
8..9	63:0	<b>Surface Base Address</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>GA63_0</td> </tr> </table> <p>Specifies the byte-aligned base address of the surface.</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td><b>Programming Notes</b></td> </tr> <tr> <td> <ul style="list-style-type: none"> <li>• 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).</li> <li>• 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.</li> <li>• Linear depth buffer surface base addresses must be 64-byte aligned. Note that while render targets (color) can be SURFTYPE_BUFFER, depth buffers cannot.</li> <li>• 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.</li> <li>• 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.</li> <li>• Other linear (non-tiled) surfaces have no alignment requirements (byte alignment is sufficient).</li> <li>• 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).</li> <li>• Certain message types used to access surfaces have more stringent alignment requirements. Please refer to the specific data-port message documentation for additional restrictions.</li> </ul> </td> </tr> </table>	Format:	GA63_0	<b>Programming Notes</b>	<ul style="list-style-type: none"> <li>• 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).</li> <li>• 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.</li> <li>• Linear depth buffer surface base addresses must be 64-byte aligned. Note that while render targets (color) can be SURFTYPE_BUFFER, depth buffers cannot.</li> <li>• 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.</li> <li>• 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.</li> <li>• Other linear (non-tiled) surfaces have no alignment requirements (byte alignment is sufficient).</li> <li>• 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).</li> <li>• Certain message types used to access surfaces have more stringent alignment requirements. Please refer to the specific data-port message documentation for additional restrictions.</li> </ul>				
Format:	GA63_0									
<b>Programming Notes</b>										
<ul style="list-style-type: none"> <li>• 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).</li> <li>• 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.</li> <li>• Linear depth buffer surface base addresses must be 64-byte aligned. Note that while render targets (color) can be SURFTYPE_BUFFER, depth buffers cannot.</li> <li>• 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.</li> <li>• 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.</li> <li>• Other linear (non-tiled) surfaces have no alignment requirements (byte alignment is sufficient).</li> <li>• 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).</li> <li>• Certain message types used to access surfaces have more stringent alignment requirements. Please refer to the specific data-port message documentation for additional restrictions.</li> </ul>										

<b>RENDER_SURFACE_STATE</b>					
	<p>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.</p>				
10..11	<p><b>63:1 Auxiliary Surface Base Address</b></p> <table border="1"> <tr> <td>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]			
<p><b>61:4 X Offset for V Plane</b></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 <b>Surface Base Address</b> to the start (origin) of the V plane.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>This field must be a multiple of 4 (bits 1:0 MBZ).</p> <p>If <b>Tiled Resource Mode</b> is enabled, this field must be a multiple of the tile width in pixels.</p> <p>This field is ignored if <b>Separate UV Plane Enable</b> is disabled.</p>	Exists If:	([Surface Format] == 'PLANAR')	Format:	U14	
Exists If:	([Surface Format] == 'PLANAR')				
Format:	U14				
<p><b>45:3 Y Offset for V Plane</b></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 vertical offset in rows from the <b>Surface Base Address</b> to the start (origin) of the V plane.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>For surfaces where <b>Surface Format</b> = PLANAR* and <b>Separate UV Plane</b> is Enabled, the Y Offset must be programmed in multiples of <b>half-rows</b>. 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).</p> <p>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.</p> <p>If <b>Tiled Resource Mode</b> is enabled, this field must be a multiple of the tile height in rows.</p> <p>This field is ignored if <b>Separate UV Plane Enable</b> is disabled.</p>	Exists If:	([Surface Format] == 'PLANAR')	Format:	U14	
Exists If:	([Surface Format] == 'PLANAR')				
Format:	U14				
11	<p><b>Procedural Texture</b></p> <p>This bit, when set, indicates that the associated surface is a procedural texture which is used for AMFS.</p> <p>This bit can be ENABLED for the following surface types: SURFTYPE_2D arrayed / non-arrayed, SURFTYPE_3D non-arrayed, SURFTYPE_CUBE arrayed/ non arrayed, and surfstype = NULL. This bit can be set for the pixel formats that are supported has typed UAVs as per the DX spec. Therefore,</p>				

<b>RENDER_SURFACE_STATE</b>																		
	<p>writes from only HDC are supported to Procedural Textures. This bit cannot be ENABLED for the following surface types: SURFTYPE_3D arrayed, SURFTYPE_BUFFER</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">This bit cannot be set when surface walk (tiling mode) is legacy Y</td> </tr> <tr> <td colspan="2">This bit cannot be set when Tiled Resource Mode = TileYS and LOD &gt;= MIP tail LOD</td> </tr> </table>	Programming Notes		This bit cannot be set when surface walk (tiling mode) is legacy Y		This bit cannot be set when Tiled Resource Mode = TileYS and LOD >= MIP tail LOD												
Programming Notes																		
This bit cannot be set when surface walk (tiling mode) is legacy Y																		
This bit cannot be set when Tiled Resource Mode = TileYS and LOD >= MIP tail LOD																		
10	<p><b>Clear Value Address Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enables HW Managed Clear Value Layout for the Surface State. If this bit is enabled, Clear Value Address is present instead of explicit clear values.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable <b>[Default]</b></td> <td>Clear values are present in the surface state explicitly.</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Clear value Address is present instead of explicit clear values.</td> </tr> </tbody> </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">Clear values can only be enabled for sampled surface formats which are supported for clear in the Pixel Data Port. See Render Target Surfaces section of <b>Pixel Data Port</b> for a list of surface types supported .</td> </tr> <tr> <td colspan="2">This bit has to be programmed to 1 if clear buffer is attached to the surface or if AUX_MODE is AUX_CCS_E. No support for explicit clear values. Only hw managed clear values are supported.</td> </tr> </table>	Format:	Enable	Value	Name	Description	0h	Disable <b>[Default]</b>	Clear values are present in the surface state explicitly.	1h	Enable	Clear value Address is present instead of explicit clear values.	Programming Notes		Clear values can only be enabled for sampled surface formats which are supported for clear in the Pixel Data Port. See Render Target Surfaces section of <b>Pixel Data Port</b> for a list of surface types supported .		This bit has to be programmed to 1 if clear buffer is attached to the surface or if AUX_MODE is AUX_CCS_E. No support for explicit clear values. Only hw managed clear values are supported.	
Format:	Enable																	
Value	Name	Description																
0h	Disable <b>[Default]</b>	Clear values are present in the surface state explicitly.																
1h	Enable	Clear value Address is present instead of explicit clear values.																
Programming Notes																		
Clear values can only be enabled for sampled surface formats which are supported for clear in the Pixel Data Port. See Render Target Surfaces section of <b>Pixel Data Port</b> for a list of surface types supported .																		
This bit has to be programmed to 1 if clear buffer is attached to the surface or if AUX_MODE is AUX_CCS_E. No support for explicit clear values. Only hw managed clear values are supported.																		
9:5	<p><b>Quilt Height</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">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%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 25%;">Name</th> <th style="width: 60%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0,31]</td> <td></td> <td>representing height of quilt - 1 (y/v dimension)</td> </tr> </tbody> </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"> <p><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>• Only power-of-2 <b>Quilt Height</b> and <b>Quilt Width</b> values are allowed: (1,2,4,8,16,32) mapping to (0,1,3,7,15,31) values in the fields.</li> <li>• A surface is defined as a quilted texture if either <b>Quilt Height</b> or <b>Quilt Width</b> is nonzero (actual field value, not the incremented value).</li> <li>• A quilted texture                             <ul style="list-style-type: none"> <li>• is only supported by the sampling engine (other shared functions will ignore the <b>Quilt Width</b> and <b>Quilt Height</b> field, behaving as if they are set to zero).</li> <li>• must have a <b>Surface Type</b> of SURFTYPE_2D.</li> <li>• must have <b>Number of Multisamples</b> set to NUMSAMPLES_1.</li> <li>• must have <b>Vertical Line Stride</b> set to 0.</li> <li>• must have <b>Auxiliary Surface Mode</b> set to AUX_NONE.</li> </ul> </li> </ul> </td> </tr> </table>	Format:	U5	Value	Name	Description	[0,31]		representing height of quilt - 1 (y/v dimension)	Programming Notes		<p><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>• Only power-of-2 <b>Quilt Height</b> and <b>Quilt Width</b> values are allowed: (1,2,4,8,16,32) mapping to (0,1,3,7,15,31) values in the fields.</li> <li>• A surface is defined as a quilted texture if either <b>Quilt Height</b> or <b>Quilt Width</b> is nonzero (actual field value, not the incremented value).</li> <li>• A quilted texture                             <ul style="list-style-type: none"> <li>• is only supported by the sampling engine (other shared functions will ignore the <b>Quilt Width</b> and <b>Quilt Height</b> field, behaving as if they are set to zero).</li> <li>• must have a <b>Surface Type</b> of SURFTYPE_2D.</li> <li>• must have <b>Number of Multisamples</b> set to NUMSAMPLES_1.</li> <li>• must have <b>Vertical Line Stride</b> set to 0.</li> <li>• must have <b>Auxiliary Surface Mode</b> set to AUX_NONE.</li> </ul> </li> </ul>						
Format:	U5																	
Value	Name	Description																
[0,31]		representing height of quilt - 1 (y/v dimension)																
Programming Notes																		
<p><b>Programming Notes</b></p> <ul style="list-style-type: none"> <li>• Only power-of-2 <b>Quilt Height</b> and <b>Quilt Width</b> values are allowed: (1,2,4,8,16,32) mapping to (0,1,3,7,15,31) values in the fields.</li> <li>• A surface is defined as a quilted texture if either <b>Quilt Height</b> or <b>Quilt Width</b> is nonzero (actual field value, not the incremented value).</li> <li>• A quilted texture                             <ul style="list-style-type: none"> <li>• is only supported by the sampling engine (other shared functions will ignore the <b>Quilt Width</b> and <b>Quilt Height</b> field, behaving as if they are set to zero).</li> <li>• must have a <b>Surface Type</b> of SURFTYPE_2D.</li> <li>• must have <b>Number of Multisamples</b> set to NUMSAMPLES_1.</li> <li>• must have <b>Vertical Line Stride</b> set to 0.</li> <li>• must have <b>Auxiliary Surface Mode</b> set to AUX_NONE.</li> </ul> </li> </ul>																		

## RENDER\_SURFACE\_STATE

			<ul style="list-style-type: none"> <li><b>Depth</b> indicates the array dimension of the quilted texture if <b>Surface Array</b> is enabled. The valid range of <b>Depth</b> is <math>[0, 2048 / (\text{QuiltWidth} * \text{QuiltHeight}) - 1]</math>, i.e. the total number of underlying array slices including quilt slices cannot exceed 2048.</li> <li>cannot be accessed with any <code>Id*</code> message type or using a sampler with the <b>Non-Normalized Coordinate Enable</b> field enabled.</li> </ul>								
			Quilted surfaces are not supported and this field must be programmed to 0h								
	4:0	<b>Quilt Width</b>	<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 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" 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 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)									
			Quilted surfaces are not supported and this field must be programmed to 0h								
12	31:6	<b>Clear Address Low</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%;">Format:</td> <td style="width: 60%;">GraphicsAddress[31:6]</td> </tr> </table>	Format:	GraphicsAddress[31:6]						
Format:	GraphicsAddress[31:6]										
			<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 100%;">Description</th> </tr> </thead> <tbody> <tr> <td> <p><b>For Sampling Engine Surfaces and Render Targets:</b> Specifies the lower bits of Graphics Address where clear value is stored in</p> <p>The memory layout of the clear color pointed to by this address is a value stored in the lower-order bytes of a 64-byte cache-line.</p> <p>The clear color will be formatted as 32-bit IEEE Floating-point per channel, 32-bit UINT per channel, 32-bit SINT per channel, or SRGB depending on the surface type (e.g. R32G32B32A32_UINT surfaces assume use 32-bit UINT for clear color). 3D Sampler will always fetch clear color from the location 16-bytes above this address, where the clear color, converted to native surface format, will be stored.</p> <p><b>For Sampling Engine Surfaces and Render Targets with Depth Surfaces:</b> Specifies the lower bits of Graphics Address where the depth clear value is stored.</p> <p>The memory format is IEEE 32 bit float. The numeric range is required to match the numeric range limitations of 3DSTATE_CLEAR_PARAMS:Depth Clear Value.</p> <p>3D Sampler will always fetch clear depth from the location 16-bytes above this address, where the clear depth, converted to native surface format by software, will be stored.</p> <p>For D24X8 depth surfaces (R24_UNORM_X8_TYPELESS), the format of the data at this location shall be UNORM24_X8 rather than a 32-bit format.</p> </td> </tr> </tbody> </table>	Description	<p><b>For Sampling Engine Surfaces and Render Targets:</b> Specifies the lower bits of Graphics Address where clear value is stored in</p> <p>The memory layout of the clear color pointed to by this address is a value stored in the lower-order bytes of a 64-byte cache-line.</p> <p>The clear color will be formatted as 32-bit IEEE Floating-point per channel, 32-bit UINT per channel, 32-bit SINT per channel, or SRGB depending on the surface type (e.g. R32G32B32A32_UINT surfaces assume use 32-bit UINT for clear color). 3D Sampler will always fetch clear color from the location 16-bytes above this address, where the clear color, converted to native surface format, will be stored.</p> <p><b>For Sampling Engine Surfaces and Render Targets with Depth Surfaces:</b> Specifies the lower bits of Graphics Address where the depth clear value is stored.</p> <p>The memory format is IEEE 32 bit float. The numeric range is required to match the numeric range limitations of 3DSTATE_CLEAR_PARAMS:Depth Clear Value.</p> <p>3D Sampler will always fetch clear depth from the location 16-bytes above this address, where the clear depth, converted to native surface format by software, will be stored.</p> <p>For D24X8 depth surfaces (R24_UNORM_X8_TYPELESS), the format of the data at this location shall be UNORM24_X8 rather than a 32-bit format.</p>						
Description											
<p><b>For Sampling Engine Surfaces and Render Targets:</b> Specifies the lower bits of Graphics Address where clear value is stored in</p> <p>The memory layout of the clear color pointed to by this address is a value stored in the lower-order bytes of a 64-byte cache-line.</p> <p>The clear color will be formatted as 32-bit IEEE Floating-point per channel, 32-bit UINT per channel, 32-bit SINT per channel, or SRGB depending on the surface type (e.g. R32G32B32A32_UINT surfaces assume use 32-bit UINT for clear color). 3D Sampler will always fetch clear color from the location 16-bytes above this address, where the clear color, converted to native surface format, will be stored.</p> <p><b>For Sampling Engine Surfaces and Render Targets with Depth Surfaces:</b> Specifies the lower bits of Graphics Address where the depth clear value is stored.</p> <p>The memory format is IEEE 32 bit float. The numeric range is required to match the numeric range limitations of 3DSTATE_CLEAR_PARAMS:Depth Clear Value.</p> <p>3D Sampler will always fetch clear depth from the location 16-bytes above this address, where the clear depth, converted to native surface format by software, will be stored.</p> <p>For D24X8 depth surfaces (R24_UNORM_X8_TYPELESS), the format of the data at this location shall be UNORM24_X8 rather than a 32-bit format.</p>											
	5	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO										
Format:	MBZ										

<b>RENDER_SURFACE_STATE</b>					
	4:0	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				
13	31	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
	30:1	6	<b>Reserved</b>		
			<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO				
Format:	MBZ				
15:0		<b>Clear Address High</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[47:32]</td> </tr> </table> <p>For Sampling Engine Surfaces and Render Targets with Auxiliary Surface Mode set to AUX_CCS: Specifies the higher bits of Graphics Address where clear value is stored from RGBA (R in the LSB and A in the MSB - in that order)</p> <p>For Depth Surfaces: Specifies the higher bits of Graphics Address.</p>	Format:	GraphicsAddress[47:32]	
Format:	GraphicsAddress[47:32]				
14	31:0	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				
15	31:0	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				



## Render Data Port Message Types

MT_DP_RT - Render Data Port Message Types														
Source:	EuSubFunctionRenderDataPort													
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	<b>Reserved</b>												
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
Access:	RO													
Format:	MBZ													
0	3:0	<b>Message Type</b> Specifies type of message												
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0Ch</td> <td>MT_RTW <b>[Default]</b></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>	Value	Name	Description	0Ch	MT_RTW <b>[Default]</b>	Render Target Write message	0Dh	MT_RTR	Render Target Read message	Others	Reserved	Ignored
		Value	Name	Description										
		0Ch	MT_RTW <b>[Default]</b>	Render Target Write message										
		0Dh	MT_RTR	Render Target Read message										
Others	Reserved	Ignored												



## Render Engine Interrupt Vector

RENDER_INTR_VEC - Render Engine Interrupt Vector						
Size (in bits):	16					
Default Value:	0x00000000					
DWord	Bit	Description				
0	15	<b>Catastrophic Error</b> This interrupt signals that a unrecoverable error during the engine processing. When Memory interface signals this error, the Command Streamer will stop parsing any more instructions. Scheduler is expected to reset the engine to evict the context				
	14	<b>EU Restart Interrupt</b> EU Restart Interrupt is generated by the GA fabric, and not by Render Command Streamer. GA routes this interrupt to GuC independently of Command Stream.				
	13	<b>Context Stall</b> Command streamer will generate a Context Stall interrupt when a high priority context gets stalled due to the other command streamer executing a normal priority or low priority context is "Run Alone" mode OR Command streamer will generate a Context Stall interrupt when a high priority context gets stalled while procuring run alone mode.				
	12	<b>Reserved</b> <table border="1" data-bbox="305 1087 1469 1182"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	11	<b>CS Wait On Semaphore</b>				
	10	<b>Spare 10</b>				
	9	<b>CS TR Invalid Tile Detection</b>				
	8	<b>CS Context Switch Interrupt</b>				
	7	<b>Legacy Context Per Process Page Fault Interrupt</b> <table border="1" data-bbox="305 1438 1469 1671"> <tr> <td>           This Fault interrupt is only delivered to the Host SW (not to GuC).            Fault interrupt is generated by GA fabric, not by the CS            This interrupt is for handling Legacy Page Fault. 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.            In Advanced (PRQ) Fault Interface is done through GUC interface.         </td> </tr> </table>	This Fault interrupt is only delivered to the Host SW (not to GuC). Fault interrupt is generated by GA fabric, not by the CS This interrupt is for handling Legacy Page Fault. 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. In Advanced (PRQ) Fault Interface is done through GUC interface.			
	This Fault interrupt is only delivered to the Host SW (not to GuC). Fault interrupt is generated by GA fabric, not by the CS This interrupt is for handling Legacy Page Fault. 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. In Advanced (PRQ) Fault Interface is done through GUC interface.					
	6	<b>CS Watchdog Counter Expired</b>				
	5	<b>Spare 5</b>				
	4	<b>CS PIPE_CONTROL Notify</b>				
3	<b>CS Error Interrupt</b>					
2	<b>Spare 2</b>					

**RENDER\_INTR\_VEC - Render Engine Interrupt Vector**

	1	Reserved
	0	CS MI User Interrupt



## Render Target Index Message Header Control

MHC_RT_RTI - Render Target Index Message Header Control		
Size (in bits):		32
Default Value:		0x00000000
DWord	Bit	Description
0	31:3	<b>Reserved</b>
		Access: RO
	Format: MBZ	
	2:0	<b>Render Target Index</b>
Format: U3		
Specifies the render target index that will be used to select blend state from BLEND_STATE.		

## Render Target Message Header

MH_RT - Render Target Message Header				
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	<b>Render Target Controls 0</b>		
		<table border="1"> <tr> <td>Format:</td> <td><b>MHC_RT_CO</b></td> </tr> </table> <p>Specifies controls for Render Target Write and Read messages.</p>	Format:	<b>MHC_RT_CO</b>
Format:	<b>MHC_RT_CO</b>			
0.1-0.1	31:0	<b>Color Calculator State Pointer</b>		
		<table border="1"> <tr> <td>Format:</td> <td><b>MHC_RT_CCSP</b></td> </tr> </table> <p>For Render Target Write message, specifies the HWORD-aligned GeneralStateOffset for Color State. Ignored by Render Target Read message.</p>	Format:	<b>MHC_RT_CCSP</b>
Format:	<b>MHC_RT_CCSP</b>			
0.2-0.2	31:0	<b>Render Target Index</b>		
		<table border="1"> <tr> <td>Format:</td> <td><b>MHC_RT_RTI</b></td> </tr> </table> <p>For Render Target Write message, specifies the render target index used to select blend state from BLEND_STATE. Ignored by Render Target Read message.</p>	Format:	<b>MHC_RT_RTI</b>
Format:	<b>MHC_RT_RTI</b>			
0.3-0.4	63:0	<b>Reserved</b>		
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
0.5-0.5	31:0	<b>Color Code</b>		
		<table border="1"> <tr> <td>Format:</td> <td><b>MHC_RT_CC</b></td> </tr> </table> <p>Hardware uses to track synchronizing events and free resources on thread completion.</p>	Format:	<b>MHC_RT_CC</b>
Format:	<b>MHC_RT_CC</b>			
0.6-0.7	63:0	<b>Reserved</b>		
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
1.0-1.0	31:0	<b>Reserved</b>		
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
1.1-1.1	31:0	<b>Poly 0</b>		
		<table border="1"> <tr> <td>Format:</td> <td><b>MHC_RT_POLY</b></td> </tr> </table> <p>Poly Information</p>	Format:	<b>MHC_RT_POLY</b>
Format:	<b>MHC_RT_POLY</b>			

<b>MH_RT - Render Target Message Header</b>		
1.2-1.2	31:0	<b>Subspan 0</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="text-align: center;"><b>MHC_RT_SUBSPAN</b></td> </tr> </table> <p>Upper left corner of subspan 0</p>
Format:	<b>MHC_RT_SUBSPAN</b>	
1.3-1.3	31:0	<b>Subspan 1</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="text-align: center;"><b>MHC_RT_SUBSPAN</b></td> </tr> </table> <p>Upper left corner of subspan 1</p>
Format:	<b>MHC_RT_SUBSPAN</b>	
1.4-1.4	31:0	<b>Subspan 2</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="text-align: center;"><b>MHC_RT_SUBSPAN</b></td> </tr> </table> <p>Upper left corner of subspan 2</p>
Format:	<b>MHC_RT_SUBSPAN</b>	
1.5-1.5	31:0	<b>Subspan 3</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="text-align: center;"><b>MHC_RT_SUBSPAN</b></td> </tr> </table> <p>Upper left corner of subspan 3</p>
Format:	<b>MHC_RT_SUBSPAN</b>	
1.6-1.6	31:0	<b>Poly 1</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="text-align: center;"><b>MHC_RT_POLY</b></td> </tr> </table> <p>Poly Information for second poly when dual-SIMD8 dispatch</p>
Format:	<b>MHC_RT_POLY</b>	
1.7-1.7	31:0	<b>Pixel Sample Enables</b>
		<table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Format:</td> <td style="text-align: center;"><b>MHC_RT_PSM</b></td> </tr> </table> <p>Pixel Sample Enables</p>
Format:	<b>MHC_RT_PSM</b>	

## Render Target Message Header Control

MHC_RT_C0 - Render Target Message Header Control						
Size (in bits):	32					
Default Value:	0x00000000					
DWord	Bit	Description				
0	31:15	<b>Reserved</b>				
		Access: RO				
		Format: MBZ				
	14	<b>Stencil Present to Render Target</b> Format: Enable For Render Target Write message, indicates that computed stencil is included in the message. Must be zero for Render Target Read message.				
	13	<b>Source Depth Present to Render Target</b> Format: Enable For Render Target Write Message, indicates that source depth data is included in the message. Must be zero for Render Target Read message.				
	12	<b>oMask to Render Target</b> Format: Enable 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.				
	11	<b>Source0 Alpha Present to Render Target</b> Format: Enable 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. <table border="1" data-bbox="337 1486 1464 1684"> <thead> <tr> <th colspan="2">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="2">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.</td> </tr> </tbody> </table>	Programming Notes		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.	
	Programming Notes					
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.						
10	<b>Reserved</b> Access: RO Format: MBZ					

<b>MHC_RT_C0 - Render Target Message Header Control</b>				
9:6	<p><b>Sample Index</b></p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <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		
<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO			
Format:	MBZ			



## Render Target Message Header Poly

MHC_RT_POLY - Render Target Message Header Poly			
Size (in bits):	32		
Default Value:	0x00000000		
DWord	Bit	Description	
0	31	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	30:27	<b>Viewport Index</b>	
		Format:	U4
			For Render Target Write message, specifies the index of the viewport currently being used. Range = [0,15] Ignored by Render Target Read message.
	26:16	<b>Render Target Array Index</b>	
		Format:	U11
		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 style="text-align: center;"><b>Programming Notes</b></p> <p>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.</p>	
15	<b>Front/Back Facing Polygon</b>		
	Format:	U1	
	Determines whether the polygon is front or back facing. Used by the render cache to determine which stencil test state to use.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	Front facing	All
1h	Back facing	All	
14:9	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
8:6	<b>Starting Sample Pair Index</b>		
	Format:	U3	
		Indicates the index of the first sample pair of the dispatch. Range = [0,3]	

MHC_RT_POLY - Render Target Message Header Poly		
	5:0	<b>Reserved</b>
		Access: RO
		Format: MBZ

## Replicated Pixel Render Target Data Payload Register

<b>MDPR_RGBA - Replicated Pixel Render Target Data Payload Register</b>						
Size (in bits):	256					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000					
DWord	Bit	Description				
0	31:0	<b>Red</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the value of all slots' red channel.	Format:	U32		
Format:	U32					
1	31:0	<b>Green</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the value of all slots' green channel.	Format:	U32		
Format:	U32					
2	31:0	<b>Blue</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the value of all slots' blue channel.	Format:	U32		
Format:	U32					
3	31:0	<b>Alpha</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table> Specifies the value of all slots' alpha channel.	Format:	U32		
Format:	U32					
4..7	127:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					



## Replicated SIMD16 Render Target Data Payload

<b>MDP_RTW_16REP - Replicated SIMD16 Render Target Data Payload</b>		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0.0-0.7	255:0	<b>RGBA</b> Format: <b>MDPR_RGBA</b> RGBA for all slots [15:0]

## Reversed SIMD Mode 2 Message Descriptor Control Field

<b>MDC_SM2R - Reversed SIMD Mode 2 Message Descriptor Control Field</b>											
Size (in bits):		1									
Default Value:		0x00000000									
<b>DWord</b>	<b>Bit</b>	<b>Description</b>									
0	0	<b>SIMD Mode</b>									
		Format: Boolean									
		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									



## S0A SIMD8 Render Target Data Payload

MDP_RTW_A8 - S0A SIMD8 Render Target Data Payload		
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Source 0 Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	<b>Red</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Red
2.0-2.7	255:0	<b>Green</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Green
3.0-3.7	255:0	<b>Blue</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
4.0-4.7	255:0	<b>Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha



MDP_RTW_A16 - S0A SIMD16 Render Target Data Payload		
6.0-6.7	255:0	<b>Blue[7:0]</b>
		Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
7.0-7.7	255:0	<b>Blue[15:8]</b>
		Format: <b>MDP_DW_SIMD8</b> Slots [15:8] Blue
8.0-8.7	255:0	<b>Alpha[7:0]</b>
		Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha
9.0-9.7	255:0	<b>Alpha[15:8]</b>
		Format: <b>MDP_DW_SIMD8</b> Slots [15:8] Alpha



## SAMPLER\_BORDER\_COLOR\_STATE

<b>SAMPLER_BORDER_COLOR_STATE</b>	
Size (in bits):	128
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000
<b>Description</b>	
<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"> <li>• 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.</li> <li>• 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.</li> </ul> <p>If the Texture Border Color Mode field in SAMPLER_STATE is set to DX9 and the MMIO register bit "Enable Missing Alpha Format Fix" is set in register E194h, then the interpretation of the border color format depends on the format of the surface being sampled:</p> <ol style="list-style-type: none"> <li>1. If the map format is UINT, border color is R8G8B8A8_UINT</li> <li>2. If the map format is SINT, border color is R8G8B8A8_SINT</li> <li>3. Otherwise, border color is R8G8B8A8_UNORM</li> </ol>	
<b>Programming Notes</b>	
<ul style="list-style-type: none"> <li>• DX9 mode is not supported for surfaces with more than 16 bits in any channel, other than 32-bit float formats which are supported.</li> <li>• The conditions under which this color is used depend on the <b>Surface Type</b> - 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.</li> <li>• 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.</li> <li>• MAPFILTER_MONO: The border color is ignored. Border color is fixed at a value of 0 by hardware.</li> </ul> <p>• 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</p>	

## SAMPLER\_BORDER\_COLOR\_STATE

state cache does not need to be invalidated.

If the Texture Border Color Mode field in SAMPLER\_STATE is set to DX9 and the MMIO register bit "Enable Missing Alpha Format Fix" is set in register E194h and a surface with format SINT is being sampled, then each channel of the border color must be 00h or 01h (0 or 1 in SINT encoding).

DWord	Bit	Description				
0	31:24	<p><b>Border Color Alpha</b></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>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><b>Border Color Red - (DX10/OGL)</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL'</td> </tr> <tr> <td>Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>Texture Border Color Mode = DX10/OGL</p>	Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL'	Format:	IEEE_FLOAT
	Exists If:	Structure[SAMPLER_STATE][Texture Border Color Mode] == 'DX10/OGL'				
Format:	IEEE_FLOAT					
23:16	<p><b>Border Color Blue</b></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>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><b>Border Color Green</b></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>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					
7:0	<p><b>Border Color Red - (DX9)</b></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>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><b>Border Color Green</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>Texture Border Color Mode = DX10/OGL</p>	Format:	IEEE_FLOAT		
Format:	IEEE_FLOAT					
2	31:0	<p><b>Border Color Blue</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>Texture Border Color Mode = DX10/OGL</p>	Format:	IEEE_FLOAT		
Format:	IEEE_FLOAT					

<b>SAMPLER_BORDER_COLOR_STATE</b>				
3	31:0	<b>Border Color Alpha</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> Texture Border Color Mode = DX10/OpenGL	Format:	IEEE_FLOAT
Format:	IEEE_FLOAT			

## SAMPLER\_INDIRECT\_STATE\_BORDER\_COLOR

<b>SAMPLER_INDIRECT_STATE_BORDER_COLOR</b>		
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
Description		
<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 <b>Indirect State Pointer</b> field in SAMPLER_STATE. The interpretation of the border color depends on the <b>Texture Border Color Mode</b> field in SAMPLER_STATE as follows:</p> <ul style="list-style-type: none"> <li>In <b>8BIT</b> 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>.</li> <li>In <b>OGL</b> 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.</li> </ul>		
Programming Notes		
<ul style="list-style-type: none"> <li>8BIT mode is not supported for surfaces with more than 16 bits in any channel, other than 32-bit float formats which are supported.</li> <li>The conditions under which this color is used depend on the <b>Surface Type</b> - 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.</li> <li>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.</li> <li>MAPFILTER_MONO: The border color is ignored. Border color is fixed at a value of 0 by hardware.</li> </ul>		
DWord	Bit	Description
0	31:24	<b>Border Color Alpha As U8</b>
		Exists If: //Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'
	Format: U8	
	31:0	<b>Border Color Red As S31</b>
Exists If: //Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true'		
Format: S31		

<b>SAMPLER_INDIRECT_STATE_BORDER_COLOR</b>											
	<table border="1"> <tr> <td>Format:</td> <td>U32</td> </tr> <tr> <td>Format:</td> <td>IEEE_FLOAT</td> </tr> </table>	Format:	U32	Format:	IEEE_FLOAT						
Format:	U32										
Format:	IEEE_FLOAT										
	<table border="1"> <tr> <td>23:16</td> <td><b>Border Color Blue As U8</b></td> </tr> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table>	23:16	<b>Border Color Blue As U8</b>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'	Format:	U8				
23:16	<b>Border Color Blue As U8</b>										
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'										
Format:	U8										
	<table border="1"> <tr> <td>15:8</td> <td><b>Border Color Green As U8</b></td> </tr> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table>	15:8	<b>Border Color Green As U8</b>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'	Format:	U8				
15:8	<b>Border Color Green As U8</b>										
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'										
Format:	U8										
	<table border="1"> <tr> <td>7:0</td> <td><b>Border Color Red As U8</b></td> </tr> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table>	7:0	<b>Border Color Red As U8</b>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'	Format:	U8				
7:0	<b>Border Color Red As U8</b>										
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'										
Format:	U8										
1	<table border="1"> <tr> <td>31:0</td> <td><b>Border Color Green As S31</b></td> </tr> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> <tr> <td>Format:</td> <td>IEEE_FLOAT</td> </tr> </table>	31:0	<b>Border Color Green As S31</b>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')	Format:	S31	Format:	U32	Format:	IEEE_FLOAT
31:0	<b>Border Color Green As S31</b>										
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')										
Format:	S31										
Format:	U32										
Format:	IEEE_FLOAT										
2	<table border="1"> <tr> <td>31:0</td> <td><b>Border Color Blue As S31</b></td> </tr> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> <tr> <td>Format:</td> <td>IEEE_FLOAT</td> </tr> </table>	31:0	<b>Border Color Blue As S31</b>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')	Format:	S31	Format:	U32	Format:	IEEE_FLOAT
31:0	<b>Border Color Blue As S31</b>										
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')										
Format:	S31										
Format:	U32										
Format:	IEEE_FLOAT										
3	<table border="1"> <tr> <td>31:0</td> <td><b>Border Color Alpha As S31</b></td> </tr> <tr> <td>Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> <tr> <td>Format:</td> <td>IEEE_FLOAT</td> </tr> </table>	31:0	<b>Border Color Alpha As S31</b>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')	Format:	S31	Format:	U32	Format:	IEEE_FLOAT
31:0	<b>Border Color Alpha As S31</b>										
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == 'OGL' AND (Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]== 'true')										
Format:	S31										
Format:	U32										
Format:	IEEE_FLOAT										

## SAMPLER\_INDIRECT\_STATE

<b>SAMPLER_INDIRECT_STATE</b>	
Size (in bits):	512
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000
<b>Description</b>	
<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 <b>Indirect State Pointer</b> (SAMPLER_STATE).</p> <p>The interpretation of the border color depends on the <b>Texture Border Color Mode</b> field in SAMPLER_STATE as follows:</p> <ul style="list-style-type: none"> <li>In <b>8BIT</b> 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>.</li> <li>In <b>OGL</b> 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.</li> </ul> <p>The format of this state depends on the <b>Texture Border Color Mode</b> field.</p>	
<b>Programming Notes</b>	
<ul style="list-style-type: none"> <li>8BIT mode is not supported for surfaces with more than 16 bits in any channel, other than 32-bit float formats which are supported.</li> <li>The conditions under which this color is used depend on the <b>Surface Type</b> - 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.</li> <li>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.</li> <li>MAPFILTER_MONO: The border color is ignored. Border color is fixed at a value of 0 by hardware.</li> </ul>	
<ul style="list-style-type: none"> <li>The conditions under which this color is used depend on the <b>Surface Type</b>- 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.</li> <li>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</li> </ul>	

## SAMPLER\_INDIRECT\_STATE

state cache does not need to be invalidated.

DWord	Bit	Description								
0	31:24	<b>Border Color Alpha</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> <p>Texture Border Color Mode = 8BIT</p>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'	Format:	UNORM8				
	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'								
	Format:	UNORM8								
	31:0	<b>Border Color Red</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>//Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned] == 'true'</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_FLOAT for all other surface formats</td> </tr> </table>	Exists If:	//Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned] == 'true'	Format:	S31 (2's complement) for all SINT surface formats	Format:	U32 for all UINT surface formats	Format:	IEEE_FLOAT for all other surface formats
	Exists If:	//Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned] == 'true'								
Format:	S31 (2's complement) for all SINT surface formats									
Format:	U32 for all UINT surface formats									
Format:	IEEE_FLOAT for all other surface formats									
23:16	<b>Border Color Blue</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> <p>Texture Border Color Mode = 8BIT</p>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'	Format:	UNORM8					
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'									
Format:	UNORM8									
15:8	<b>Border Color Green</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> <p>Texture Border Color Mode = 8BIT</p>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'	Format:	UNORM8					
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'									
Format:	UNORM8									
7:0	<b>Border Color Red</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'</td> </tr> <tr> <td>Format:</td> <td>UNORM8</td> </tr> </table> <p>Texture Border Color Mode = 8BIT</p>	Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'	Format:	UNORM8					
Exists If:	//Structure[SAMPLER_STATE][Texture Border Color Mode] == '8BIT'									
Format:	UNORM8									
1	31:0	<b>Border Color Green</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>//Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned] == 'true'</td> </tr> <tr> <td>Format:</td> <td>IEEE_FLOAT</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table>	Exists If:	//Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned] == 'true'	Format:	IEEE_FLOAT	Format:	S31	Format:	U32
		Exists If:	//Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned] == 'true'							
		Format:	IEEE_FLOAT							
		Format:	S31							
Format:	U32									
2	31:0	<b>Border Color Blue</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 15%;">Exists If:</td> <td>//Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned] == 'true'</td> </tr> <tr> <td>Format:</td> <td>IEEE_FLOAT</td> </tr> <tr> <td>Format:</td> <td>S31</td> </tr> <tr> <td>Format:</td> <td>U32</td> </tr> </table>	Exists If:	//Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned] == 'true'	Format:	IEEE_FLOAT	Format:	S31	Format:	U32
		Exists If:	//Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned] == 'true'							
		Format:	IEEE_FLOAT							
Format:	S31									
Format:	U32									

<b>SAMPLER_INDIRECT_STATE</b>		
3	31:0	<b>Border Color Alpha</b>
		Exists If: //Structure[RENDER_SURFACE_STATE][Surface Format]Property[IsSigned]='true'
		Format: IEEE_FLOAT
		Format: U32
4..15	383:0	<b>Reserved</b>
		Access: RO
		Format: MBZ





<b>SAMPLER_STATE_8x8_AVS</b>																	
0x00000000, 0x00000000																	
Description																	
ExistsIf = AVS																	
DWord	Bit	Description															
0..2	95:0	<b>Reserved</b>															
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ											
Access:	RO																
Format:	MBZ																
3	31:30	<b>Reserved</b>															
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ											
Access:	RO																
Format:	MBZ																
	29:28	<b>Enable 8-tap filter</b> <b>Adaptive Filtering (Mode = 11) ExistsIf:</b> R10G10B10A2_UNORM R8G8B8A8_UNORM (AYUV also) R8B8G8A8_UNORM B8G8R8A8_UNORM R16G16B16A16 <i>Enable 8-tap Filtering on UV channel (Mode = 10) ExistsIf:</i> R16B16_UNORM, R16_UNORM <b>Enable 8-tap Filtering on UV channel (Mode = 10) ExistsIf:</b> R10G10B10A2_UNORMR8G8B8A8_UNORM (AYUV also)R8B8_UNORM (CrCb)R8_UNORMR8B8G8A8_UNORMB8G8R8A8_UNORMR16G16B16A16Y8_UNORM															
		<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td></td> <td>4-tap filter is only done on all channels.</td> </tr> <tr> <td>01b</td> <td></td> <td>Enable 8-tap Adaptive filter on G-channel. 4-tap filter on other channels.</td> </tr> <tr> <td>10b</td> <td></td> <td>8-tap filter is done on all channels (UV-ch uses the Y-coefficients)</td> </tr> <tr> <td>11b</td> <td></td> <td>Enable 8-tap Adaptive filter all channels (UV-ch uses the Y-coefficients).</td> </tr> </tbody> </table>	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).
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).															
		<b>Programming Notes</b>															
		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															

<b>SAMPLER_STATE_8x8_AVS</b>													
	27:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ							
Access:	RO												
Format:	MBZ												
4	31:12	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ							
	Access:	RO											
	Format:	MBZ											
	11	<b>Shuffle_OutputWriteback for sample_8x8</b> <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	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
Access:	RO												
Format:	MBZ												
5..15	351:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ							
Access:	RO												
Format:	MBZ												
16..151	4351:0	<b>Filter Coefficient[0..16]</b> <table border="1"> <tr> <td>Format:</td> <td><b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS[17]</b></td> </tr> </table>	Format:	<b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS[17]</b>									
Format:	<b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS[17]</b>												
152	31:24	<b>Default Sharpness Level</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>When adaptive scaling is off, determines the balance between sharp and smooth scalars.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td><b>[Default]</b></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>	Format:	U8	Value	Name	Description	0	<b>[Default]</b>	Contribute 1 from the smooth scalar	255		Contribute 1 from the sharp scalar
	Format:	U8											
	Value	Name	Description										
	0	<b>[Default]</b>	Contribute 1 from the smooth scalar										
	255		Contribute 1 from the sharp scalar										
23:16	<b>Max Derivative 4 Pixels</b> <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 4 pixel area.</p>	Format:	U8										
Format:	U8												
15:8	<b>Max Derivative 8 Pixels</b> <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	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
Access:	RO												
Format:	MBZ												

<b>SAMPLER_STATE_8x8_AVS</b>													
	6:4	<b>Transition Area with 4 Pixels</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">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	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ							
Access:	RO												
Format:	MBZ												
2:0	<b>Transition Area with 8 Pixels</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">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	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ							
	Access:	RO											
	Format:	MBZ											
	22	<b>Bypass X Adaptive Filtering</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Disable</td> </tr> </table> <p>When disabled, the X direction will use <b>Default Sharpness Level</b> to blend between the smooth and sharp filters rather than the calculated value.</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>1</td> <td>Disble</td> <td>Disable X Adaptive 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 Adaptive Filtering	0	Enable	Enable X Adaptive Filtering
	Format:	Disable											
Value	Name	Description											
1	Disble	Disable X Adaptive Filtering											
0	Enable	Enable X Adaptive Filtering											
21	<b>Bypass Y Adaptive Filtering</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Disable</td> </tr> </table> <p>When disabled, the Y direction will use <b>Default Sharpness Level</b> to blend between the smooth and sharp filters rather than the calculated value.</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>1</td> <td>Disble</td> <td>Disable Y Adaptive 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 Adaptive Filtering	0	Enable	Enable Y Adaptive Filtering	
Format:	Disable												
Value	Name	Description											
1	Disble	Disable Y Adaptive Filtering											
0	Enable	Enable Y Adaptive Filtering											
20:2	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ								
Access:	RO												
Format:	MBZ												
1	<b>Adaptive Filter for all channels</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Only to be enabled if 8-tap Adaptive filter mode is on, eElse it should be disabled.</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>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>	Format:	Enable	Value	Name	Description	1	Enable	Enable Adaptive Filter on UV/RB Channels	0	Disble	Disable Adaptive Filter on UV/RB Channels	
Format:	Enable												
Value	Name	Description											
1	Enable	Enable Adaptive Filter on UV/RB Channels											
0	Disble	Disable Adaptive Filter on UV/RB Channels											

<b>SAMPLER_STATE_8x8_AVS</b>													
	0	<p><b>RGB Adaptive</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>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.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Enable</td> <td>Enable the RGB Adaptive filter using the equation <math>(Y=(R+2G+B)\gg 2)</math></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>	Format:	Enable	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
Format:	Enable												
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	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ							
Access:	RO												
Format:	MBZ												
160..279	3839:0	<p><b>Filter Coefficient[17..31]</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS[15]</b></td> </tr> </table>	Format:	<b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS[15]</b>									
Format:	<b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS[15]</b>												



## SAMPLER\_STATE\_8x8\_AVS\_COEFFICIENTS

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

<b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS</b>		
	7:0	<b>Table 0X Filter Coefficient[n,2]</b>
		Format: <span style="float: right;">S1.6</span> <b>Range:</b> [-2.0, +2.0)
2	31:24	<b>Table 0Y Filter Coefficient[n,5]</b>
		Format: <span style="float: right;">S1.6</span> <b>Range:</b> [-2.0, +2.0)
	23:16	<b>Table 0X Filter Coefficient[n,5]</b>
		Format: <span style="float: right;">S1.6</span> <b>Range:</b> [-2.0, +2.0)
	15:8	<b>Table 0Y Filter Coefficient[n,4]</b>
		Format: <span style="float: right;">S1.6</span> <b>Range:</b> [-2.0, +2.0)
		<b>Programming Notes</b>
7:0	<b>Table 0X Filter Coefficient[n,4]</b>	
	Format: <span style="float: right;">S1.6</span> <b>Range:</b> [-2.0, +2.0)	
	<b>Programming Notes</b>	
		If the format is R10G10B10A2_UNORM or R8G8B8A8_UNORM, this field MBZ.
3	31:24	<b>Table 0Y Filter Coefficient[n,7]</b>
		Format: <span style="float: right;">S1.6</span> <b>Range:</b> [-2, +2)
	23:16	<b>Table 0X Filter Coefficient[n,7]</b>
		Format: <span style="float: right;">S1.6</span> <b>Range:</b> [-2, +2)
	15:8	<b>Table 0Y Filter Coefficient[n,6]</b>
		Format: <span style="float: right;">S1.6</span> <b>Range:</b> [-2, +2)
	7:0	<b>Table 0X Filter Coefficient[n,6]</b>
		Format: <span style="float: right;">S1.6</span> <b>Range:</b> [-2, +2)

<b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS</b>						
4	31:24	<b>Table 1X Filter Coefficient[n,3]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S1.6</td> </tr> </table> <b>Range:</b> [-2.0, +2.0)	Format:	S1.6		
	Format:	S1.6				
	23:16	<b>Table 1X Filter Coefficient[n,2]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S1.6</td> </tr> </table> <b>Range:</b> [-2.0, +2.0)	Format:	S1.6		
Format:	S1.6					
15:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
5	31:16	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
15:8	<b>Table 1X Filter Coefficient[n,5]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S1.6</td> </tr> </table> <b>Range:</b> [-2.0, +2.0)	Format:	S1.6			
Format:	S1.6					
7:0	<b>Table 1X Filter Coefficient[n,4]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S1.6</td> </tr> </table> <b>Range:</b> [-2.0, +2.0)	Format:	S1.6			
Format:	S1.6					
6	31:24	<b>Table 1Y Filter Coefficient[n,3]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S1.6</td> </tr> </table> <b>Range:</b> [-2.0, +2.0)	Format:	S1.6		
	Format:	S1.6				
	23:16	<b>Table 1Y Filter Coefficient[n,2]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S1.6</td> </tr> </table> <b>Range:</b> [-2.0, +2.0)	Format:	S1.6		
Format:	S1.6					
15:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
7	31:16	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
15:8	<b>Table 1Y Filter Coefficient[n,5]</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>S1.6</td> </tr> </table> <b>Range:</b> [-2.0, +2.0)	Format:	S1.6			
Format:	S1.6					



<b>SAMPLER_STATE_8x8_AVS_COEFFICIENTS</b>				
	7:0	<b>Table 1Y Filter Coefficient[n,4]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>S1.6</td> </tr> </table> <b>Range:</b> [-2.0, +2.0)	Format:	S1.6
Format:	S1.6			

## SAMPLER\_STATE

<b>SAMPLER_STATE</b>																											
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	<p><b>Sampler Disable</b></p> <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	<p><b>CPS LOD Compensation Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field, if enabled, causes derivatives used to compute LOD to be adjusted by scale factors for coarse pixel shading. The adjustment only occurs if the following are all true:</p> <ul style="list-style-type: none"> <li>This field is enabled</li> <li><b>CPS Message LOD Compensation Enable</b> in the message header is enabled</li> </ul> <p>The scale.x and scale.y factors are computed in hardware and delivered to the sampler at thread dispatch time.</p> <p>The following adjustments generate new derivatives as follows:</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td><math>\frac{du}{dx}</math></td> <td>=</td> <td><math>\frac{du}{dx}</math></td> <td>*scale.x</td> <td><math>\frac{dv}{dx}</math></td> <td>=</td> <td><math>\frac{dv}{dx}</math></td> <td>*scale.x</td> <td><math>\frac{dr}{dx}</math></td> <td>=</td> <td><math>\frac{dr}{dx}</math></td> <td>*scale.x</td> </tr> <tr> <td><math>\frac{du}{dy}</math></td> <td>=</td> <td><math>\frac{du}{dy}</math></td> <td>*scale.y</td> <td><math>\frac{dv}{dy}</math></td> <td>=</td> <td><math>\frac{dv}{dy}</math></td> <td>*scale.y</td> <td><math>\frac{dr}{dy}</math></td> <td>=</td> <td><math>\frac{dr}{dy}</math></td> <td>*scale.y</td> </tr> </table>	Format:	Enable	$\frac{du}{dx}$	=	$\frac{du}{dx}$	*scale.x	$\frac{dv}{dx}$	=	$\frac{dv}{dx}$	*scale.x	$\frac{dr}{dx}$	=	$\frac{dr}{dx}$	*scale.x	$\frac{du}{dy}$	=	$\frac{du}{dy}$	*scale.y	$\frac{dv}{dy}$	=	$\frac{dv}{dy}$	*scale.y	$\frac{dr}{dy}$	=	$\frac{dr}{dy}$
Format:	Enable																										
$\frac{du}{dx}$	=	$\frac{du}{dx}$	*scale.x	$\frac{dv}{dx}$	=	$\frac{dv}{dx}$	*scale.x	$\frac{dr}{dx}$	=	$\frac{dr}{dx}$	*scale.x																
$\frac{du}{dy}$	=	$\frac{du}{dy}$	*scale.y	$\frac{dv}{dy}$	=	$\frac{dv}{dy}$	*scale.y	$\frac{dr}{dy}$	=	$\frac{dr}{dy}$	*scale.y																
29	<p><b>Texture Border Color Mode</b></p> <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%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>OGL</td> <td>New mode for interpreting the border color</td> </tr> <tr> <td>1h</td> <td>8BIT</td> <td>Earlier mode for interpreting the border color</td> </tr> </tbody> </table>	Value	Name	Description	0h	OGL	New mode for interpreting the border color	1h	8BIT	Earlier mode for interpreting the border color																	
Value	Name	Description																									
0h	OGL	New mode for interpreting the border color																									
1h	8BIT	Earlier mode for interpreting the border color																									

## SAMPLER\_STATE

Programming Notes														
This must not be set to 8BIT 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 8BIT mode when used with surfaces that have Surface Format P4A4_UNORM or A4P4_UNORM.														
This field must be set to OGL mode when used with surfaces that have Surface Format YCRCB_SWAPUV or YCRCB_SWAPY.														
This field must be set to OGL mode if <b>Surface Format</b> for the associated surface is UINT OR SINT except when setting BORDER COLOR RED/GREEN/BLUE and ALPHA to 0														
This field must be set to OGL mode if REDUCTION_MINIMUM or REDUCTION_MAXIMUM or message type is sample_min or sample_max.														
Undefined behavior if 8BIT mode is used with any feature added. See Legacy sampler feature page for details														
28:27	<b>LOD PreClamp Mode</b>													
This field determines whether the computed LOD is clamped to [max,min] mip level before the mag-vs-min determination is performed.														
PRECLAMP_OGL: LOD pre-clamped to <b>Min LOD</b> and <b>Max LOD</b>														
OpenGL API currently clamps LOD to the <b>Min LOD</b> and <b>Max LOD</b> (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.														
<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>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	<b>Low Quality Cube Corner Mode Enable</b>													
Format:		U1												
This bit, when set to 1, forces sampler to use low-quality filtering for Cube Corners with texel replication which is not compatible with DirectX														
When cleared to 0 (default), the sampler will use a high-quality filtering for Cube Corners with 3-way texel averaging.														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 20%;">Name</th> <th style="width: 65%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disable <b>[Default]</b></td> <td>Disables low-quality Cube Corner mode</td> </tr> <tr> <td>1h</td> <td>Enable</td> <td>Enables low-quality Cube Corner mode</td> </tr> </tbody> </table>			Value	Name	Description	0h	Disable <b>[Default]</b>	Disables low-quality Cube Corner mode	1h	Enable	Enables low-quality Cube Corner mode			
Value	Name	Description												
0h	Disable <b>[Default]</b>	Disables low-quality Cube Corner mode												
1h	Enable	Enables low-quality Cube Corner mode												
25:22	<b>Reserved</b>													
Access:		RO												
Format:		MBZ												

## SAMPLER\_STATE

21:20	<b>Mip Mode Filter</b>		U2
		Format:	
<p>This field determines if and how mip map levels are chosen and/or combined when texture filtering.</p>			
<b>Value</b>	<b>Name</b>	<b>Description</b>	
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).	
<b>Programming Notes</b>			
MIPFILTER_LINEAR is not 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.			
MIP Mode Filter must be set to NONE for Planar YUV surfaces.			
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, maximum or gather4 operation is being performed.			
19:17	<b>Mag Mode Filter</b>		U3
		Format:	
<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>			
<b>Value</b>	<b>Name</b>	<b>Description</b>	
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	Reserved		
7h	Reserved		
<b>Programming Notes</b>			
Only MAPFILTER_NEAREST and MAPFILTER_LINEAR are supported for surfaces of type SURFTYPE_3D.			
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.			
MAPFILTER_ANISOTROPIC will be converted to linear if enabled for cube maps with the			

## SAMPLER\_STATE

		<p>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 are allowed with UINT/SINT if a minimum, maximum or gather4 operation is being performed.</p>																						
16:14	<b>Min Mode Filter</b>	Format:	U3																					
		<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>Reserved</td> <td></td> </tr> <tr> <td>7h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>		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	Reserved		7h	Reserved	
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	Reserved																							
7h	Reserved																							
13:1	<b>Texture LOD Bias</b>	Format:	S4.8																					
		<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>																						
		<p><b>Programming Notes</b></p>																						
		<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>																						
0	<b>LOD algorithm</b>	Format:	U1																					
		<p>Controls which algorithm is used for LOD calculation. 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>		Value	Name	Description	0h	LEGACY	Use the legacy algorithm for anisotropic filtering	1h	EWA Approximation	Use the new EWA approximation algorithm for anisotropic filtering												
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	31:20	<b>Min LOD</b>	
		Format:	U4.8
		Range: [0.0, 14.0], where the upper limit is also bounded by the Max LOD.	
		<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>	
<b>Programming Notes</b>			
If Min LOD is greater than Max LOD, Min LOD takes precedence, i.e. the resulting LOD will always be Min LOD.			
19:8		<b>Max LOD</b>	
		Format:	U4.8
		Range: [0.0, 14.0]	
		<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>	
7		<b>ChromaKey Enable</b>	
		Format:	Enable
		<b>Programming Notes</b>	
		<p>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_ANISOTROPIC. This field must be disabled if used with a surface of type SURFTYPE_3D. This field must be disabled when Mip Mode Filter is no NONE.</p> <p>This bit must not be set if the <b>Auxiliary Surface Mode</b> is not AUX_NONE.</p>	
6:5		<b>ChromaKey Index</b>	
		Format:	U2
		Range: [0, 3]	
		<p>This field specifies the index of the ChromaKey Table entry associated with this Sampler. This field is a "don't care" unless <b>ChromaKey Enable</b> is ENABLED.</p>	

## SAMPLER\_STATE

### 4 ChromaKey Mode

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

This field specifies the behavior of the device in the event of a ChromaKey match. This field is ignored if ChromaKey is disabled.

**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.

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

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

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.

Value	Name
0h	PREFILTEROP ALWAYS
1h	PREFILTEROP NEVER
2h	PREFILTEROP LESS
3h	PREFILTEROP EQUAL
4h	PREFILTEROP LEQUAL
5h	PREFILTEROP GREATER
6h	PREFILTEROP NOTEQUAL

<b>SAMPLER_STATE</b>									
	<table border="1"> <tr> <td style="width: 150px;">7h</td> <td>PREFILTEROP GEQUAL</td> </tr> </table>	7h	PREFILTEROP GEQUAL						
7h	PREFILTEROP GEQUAL								
0	<p><b>Cube Surface Control Mode</b></p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>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.</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>PROGRAMMED</td> </tr> <tr> <td>1h</td> <td>OVERRIDE</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0h	PROGRAMMED	1h	OVERRIDE
Format:	U1								
Value	Name								
0h	PROGRAMMED								
1h	OVERRIDE								
2	<p><b>31:24 Extended Indirect State Pointer</b></p> <p>These 8-bits represent the 8 msb's of the Indirect State Pointer to expand the offset from 16Mbytes to 4GBytes.</p> <p>These 8-bits in conjunction with the <b>Indirect State Pointer</b> field are the pointer to SAMPLER_INDIRECT_STATE, which contains the border color.</p> <p>The pointer is relative to the Dynamic State Base Address for Non-Bindless sampler state, and is relative to the Sample State Base Address for Bindless sampler state</p>								
	<p><b>23:6 Indirect State Pointer</b></p> <table border="1"> <tr> <td>Format:</td> <td>DynamicStateOffset[23:6]SAMPLER_INDIRECT_STATE_BORDER_COLOR</td> </tr> </table> <p>This field specifies the pointer to SAMPLER_INDIRECT_STATE, which contains the border color.</p> <p>This pointer is relative to the Dynamic State Base Address for Non-Bindless sampler state, and is relative to the Sample State Base Address for Bindless sampler state</p> <p>If a static sampler state (included in message) is being used (by setting the associated MMIO bit in SAMPLER_MODE register and setting the sampler index to 0xF), then lsb (bit 6) contains a 0 or 1 to indicate a fixed border color of black or white. The other bits of this field are ignored when static sampler state is being used.</p>	Format:	DynamicStateOffset[23:6]SAMPLER_INDIRECT_STATE_BORDER_COLOR						
Format:	DynamicStateOffset[23:6]SAMPLER_INDIRECT_STATE_BORDER_COLOR								
5	<p><b>Force gather4 Behavior</b></p> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td> <p><b>Note: This feature should not be enabled. It must remain programmed to 0h.</b></p> <p>This field, if enabled, specifies that the sampler should convert all SIMD8*, and SIMD16 sample* messages to behave as if the incoming message is a modified <i>gather4</i>, regardless of the actual message delivered. Any parameters included in the incoming message that are not needed by the gather4 operation are ignored by the sampler. The <b>Gather4 Source Channel Select</b> in the message header is ignored and set to the RED channel. The channel to sample mapping is modified from the normal <i>gather4</i> message as follows:</p> <table border="1"> <tr> <td>upper left sample = alpha channel</td> <td>upper right sample = red channel</td> </tr> <tr> <td>lower left sample = green channel</td> <td>lower right sample = blue channel</td> </tr> </table> </td> </tr> </tbody> </table>	Format:	Enable	Description	<p><b>Note: This feature should not be enabled. It must remain programmed to 0h.</b></p> <p>This field, if enabled, specifies that the sampler should convert all SIMD8*, and SIMD16 sample* messages to behave as if the incoming message is a modified <i>gather4</i>, regardless of the actual message delivered. Any parameters included in the incoming message that are not needed by the gather4 operation are ignored by the sampler. The <b>Gather4 Source Channel Select</b> in the message header is ignored and set to the RED channel. The channel to sample mapping is modified from the normal <i>gather4</i> message as follows:</p> <table border="1"> <tr> <td>upper left sample = alpha channel</td> <td>upper right sample = red channel</td> </tr> <tr> <td>lower left sample = green channel</td> <td>lower right sample = blue channel</td> </tr> </table>	upper left sample = alpha channel	upper right sample = red channel	lower left sample = green channel	lower right sample = blue channel
Format:	Enable								
Description									
<p><b>Note: This feature should not be enabled. It must remain programmed to 0h.</b></p> <p>This field, if enabled, specifies that the sampler should convert all SIMD8*, and SIMD16 sample* messages to behave as if the incoming message is a modified <i>gather4</i>, regardless of the actual message delivered. Any parameters included in the incoming message that are not needed by the gather4 operation are ignored by the sampler. The <b>Gather4 Source Channel Select</b> in the message header is ignored and set to the RED channel. The channel to sample mapping is modified from the normal <i>gather4</i> message as follows:</p> <table border="1"> <tr> <td>upper left sample = alpha channel</td> <td>upper right sample = red channel</td> </tr> <tr> <td>lower left sample = green channel</td> <td>lower right sample = blue channel</td> </tr> </table>	upper left sample = alpha channel	upper right sample = red channel	lower left sample = green channel	lower right sample = blue channel					
upper left sample = alpha channel	upper right sample = red channel								
lower left sample = green channel	lower right sample = blue channel								



## SAMPLER\_STATE

4	<b>Reserved</b>		
	Access:		RO
	Format:		MBZ
3	<b>Return Filter Weight for Border Texels</b>	<p>This bit, when set, returns the filter_weight in the Alpha channel of all non-border texels. Red, Green, and Blue channels will contain the sample result with border texels excluded. For cases where the surface format contains an Alpha channel, the result returned will be overwritten to return the filter weight. For cases where the surface format does not contain Alpha, the result will still be returned in the Alpha Channel.</p>	
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	Disable <b>[Default]</b>	When programmed to 0h, normal data will be returned on RGBA channels, including contribution from border color texels.
	1h	Enable	When programmed to 1h, RGB channels return filter data contributed from non-border color texels, and A channel returns filter weight of contributing texels.
<b>Programming Notes</b>			
<p>If this bit is set then the border color and the Border Color Mode field (in SAMPLER_STATE) are ignored.</p> <p>Certain message types such as sample_c, sample_min/max and gather4_* have restrictions on the use of this mode. See the Messages section of the 3D sampler for more information.</p>			
2	<b>Return Filter Weight for Null Texels</b>	<p>This bit, when set, causes samples to return filter_weight of all non-NULL texels in the Alpha channel; Red, Green, and Blue channels are contain the filter result with NULL texels excluded; A non-NULL texel is a texel which does not reference a Null Tile. For cases where Tiled_Resource_Mode is TR_NONE, the result will always be 1.0 since no texels would be NULL. For cases where the surface format contains an Alpha channel, the result returned will be overridden to return the filter weight. For cases where the surface format does not contain Alpha, the result will still be returned in the Alpha Channel.</p>	
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0h	Disable <b>[Default]</b>	When programmed to 0h, filter weight will not be returned, and normal data will be returned on the Alpha channel.
	1h	Enable	When programmed to 1h, filter weight will be returned on the Alpha channel rather than the normal data expected on the Alpha channel.
<b>Programming Notes</b>			
<p>Certain message types such as sample_c, sample_min/max and gather4_* have restrictions on the use of this mode. See the Messages section of the 3D sampler for more information.</p>			

## SAMPLER\_STATE

1		<b>SRGB DECODE</b>	<p>This bit controls whether the 3D sampler will decode an sRGB formatted surface into RGB prior to any filtering operation.</p> <p>When set, it does not convert to linear RGB (via a reverse gamma conversion). This bit is ignored for ASTC formats, which are always converted to linear RGB prior to filtering.</p>	
		<b>Value</b>	<b>Name</b>	<b>Description</b>
		0h	DECODE_EXT <b>[Default]</b>	When set to 0h, the 3D sampler will convert texels from an sRGB surface to linear RGB prior to filtering and/or returning the value.
		1h	SKIP_DECODE_EXT	When set to 1h, the 3D sampler will not convert texels to linear RGB before filtering and returning results.
0		<b>LOD Clamp Magnification Mode</b>		
		Format:		U1
		This field allows the flexibility to control how LOD clamping is handled when in magnification mode.		
		<b>Value</b>	<b>Name</b>	<b>Description</b>
		0h	MIPNONE	When in magnification mode, Sampler will clamp LOD as if the <b>Mip Mode Filter</b> 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 <b>Mip Mode Filter</b> .
3	31:27	<b>Reserved</b>		
		Access:		RO
		Format:		MBZ
26		<b>low quality filter</b>		
		Format:		enable
		Setting this bit will enable low quality filter to save power.		
		*Will result in lower precision		
		* only has an affect if the surface format is unorm8 in the sampler L1		
		* has no affect if in anisotropic mode.		
		<b>Value</b>	<b>Name</b>	<b>Description</b>
		0h	Disable <b>[Default]</b>	When set to 0h, filter quality is high and there is no degradation in precision. Power will be higher for some surface formats.
		1h	Enable	When set to 1h, filter quality is lower and precision is reduced. Power will be lower for these surface formats.
		<b>Programming Notes</b>		
		This bit should always be programmed to 1 to ensure low power operation.		
25		<b>Reserved</b>		
		Access:		RO
		Format:		MBZ

## SAMPLER\_STATE

24 **Allow low quality LOD calculation**

Format:	enable
---------	--------

Setting this bit will allow sampler to use the low quality LOD calculation mode for power savings. Note that this will not force low quality and sampler will only do it if the follow conditions are also true. If they are not true it will use the same algorithm as before as selected by the EWA bit

Message type sample/sample\_l  
 Min/Mag/Mip\_filter = nearest or linear.  
 Map type = 2D //No arrays  
 Indirect offsets must be zero  
 Coordinates must be normalized  
 No clamp border or half border  
 Sampler must not be disabled //Sampler state bit  
 No chromakey  
 No posh

23:22 **Reduction Type**

Format:	U2
---------	----

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*.

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*,

## SAMPLER\_STATE

		not the restrictions of <i>sample_c</i> .	
21:19	<b>Maximum Anisotropy</b>		
	Format:	U3	
	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	
7h	RATIO 16:1	At most a 16:1 aspect ratio filter is used	
18	<b>U Address Mag Filter Rounding Enable</b>		
	Format:	Enable	
	Controls whether the texture address is converted to 16.8 fixed point before being truncated to select to sample. Provides independent control of rounding on one texture address dimension (U/V/R) in either mag or min filter mode.		
	<b>Programming Notes</b>		
	Hardware will <b>not</b> force rounding enable. <b>U Address Min Rounding Enable</b> and <b>U Address Mag Filter Rounding Enable</b> must be set to the same value if the <b>Min Mode Filter</b> and <b>Mag Mode Filter</b> are programmed to the same value.		
17	<b>U Address Min Filter Rounding Enable</b>		
	Format:	Enable	
	Controls whether the texture address is converted to 16.8 fixed point before being truncated to select to sample. Provides independent control of rounding on one texture address dimension (U/V/R) in either mag or min filter mode.		
	<b>Programming Notes</b>		
	Hardware will <b>not</b> force rounding enable. <b>U Address Min Rounding Enable</b> and <b>U Address Mag Filter Rounding Enable</b> must be set to the same value if the <b>Min Mode Filter</b> and <b>Mag Mode Filter</b> are programmed to the same value.		
16	<b>V Address Mag Filter Rounding Enable</b>		
	Format:	Enable	
Controls whether the texture address is converted to 16.8 fixed point before being truncated to select to sample. Provides independent control of rounding on one texture address dimension (U/V/R) in either mag or min filter mode.			

## SAMPLER\_STATE

Programming Notes		
Hardware will <b>not</b> force rounding enable.		
<b>V Address Min Rounding Enable</b> and <b>V Address Mag Filter Rounding Enable</b> must be set to the same value if the <b>Min Mode Filter</b> and <b>Mag Mode Filter</b> are programmed to the same value.		
15	<b>V Address Min Filter Rounding Enable</b>	
	Format:	Enable
Controls whether the texture address is converted to 16.8 fixed point before being truncated to select 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 <b>not</b> force rounding enable.		
<b>V Address Min Rounding Enable</b> and <b>V Address Mag Filter Rounding Enable</b> must be set to the same value if the <b>Min Mode Filter</b> and <b>Mag Mode Filter</b> are programmed to the same value.		
14	<b>R Address Mag Filter Rounding Enable</b>	
	Format:	Enable
Controls whether the texture address is converted to 16.8 fixed point before being truncated to select 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 <b>not</b> force rounding enable.		
<b>R Address Min Rounding Enable</b> and <b>R Address Mag Filter Rounding Enable</b> must be set to the same value if the <b>Min Mode Filter</b> and <b>Mag Mode Filter</b> are programmed to the same value.		
13	<b>R Address Min Filter Rounding Enable</b>	
	Format:	Enable
Controls whether the texture address is converted to 16.8 fixed point before being truncated to select 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 <b>not</b> force rounding enable.		
<b>R Address Min Rounding Enable</b> and <b>R Address Mag Filter Rounding Enable</b> must be set to the same value if the <b>Min Mode Filter</b> and <b>Mag Mode Filter</b> are programmed to the same value.		
12:11	<b>MIP Linear Filter Quality</b>	
This 2-bit field controls the rounding of LOD for MIP Linear Filtering modes (e.g. Trilinear, etc).		
Value	Name	Description
0h	FULL QUALITY <b>[Default]</b>	No rounding of LOD is done, the full 8-bit quality is used for filtering.

<b>SAMPLER_STATE</b>			
	1h	HIGH QUALITY	LOD values which are within 12.5% of an integer LOD value are rounded to that value prior to filtering and filtering effectively becomes the same as MIP Nearest.
	2h	MEDIUM QUALITY	LOD values which are within 16.67% of an integer LOD value are rounded to that value prior to filtering and filtering effectively becomes the same as MIP Nearest.
	3h	LOW QUALITY	LOD values which are within 25% of an integer LOD value are rounded to that value prior to filtering and filtering effectively becomes the same as MIP Nearest.
10	<b>Non-normalized Coordinate Enable</b>		
	Format:		Enable
	<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>		
	<b>Programming Notes</b>		
	<p>The following state must be set as indicated if this field is <i>enabled</i>:</p> <ul style="list-style-type: none"> <li>• TCX/Y/Z Address Control Mode must be TEXCOORDMODE_CLAMP, TEXCOORDMODE_HALF_BORDER, or TEXCOORDMODE_CLAMP_BORDER.</li> <li>• Surface Type must be SURFTYPE_2D or SURFTYPE_3D.</li> <li>• Mag Mode Filter must be MAPFILTER_NEAREST or MAPFILTER_LINEAR.</li> <li>• Min Mode Filter must be MAPFILTER_NEAREST or MAPFILTER_LINEAR.</li> <li>• Mip Mode Filter must be MIPFILTER_NONE.</li> <li>• Min LOD must be 0.</li> <li>• Max LOD must be 0.</li> <li>• MIP Count must be 0.</li> <li>• Surface Min LOD must be 0.</li> <li>• Texture LOD Bias must be 0.</li> </ul>		
9	<b>Reduction Type Enable</b>		
	Format:		Enable
	<p>This field enables the <b>Reduction Type</b> field to modify the behavior of messages based on its setting. If this field is disabled, all messages behave as defined and the <b>Reduction Type</b> field is ignored.</p>		
8:6	<b>TCX Address Control Mode</b>		
	Format:	<b>Texture Coordinate Mode</b>	
	<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>		

<b>SAMPLER_STATE</b>											
	<table border="1"> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> <tr> <td colspan="2">When using cube map texture coordinates, each TC component must have the same Address Control Mode.</td> </tr> <tr> <td colspan="2">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).</td> </tr> <tr> <td colspan="2">If <b>Surface Format</b> is PLANAR*, this field must be set to TEXCOORDMODE_CLAMP.</td> </tr> </table>	<b>Programming Notes</b>		When using cube map texture coordinates, each TC component must have the same Address Control Mode.		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).		If <b>Surface Format</b> is PLANAR*, this field must be set to TEXCOORDMODE_CLAMP.			
<b>Programming Notes</b>											
When using cube map texture coordinates, each TC component must have the same Address Control Mode.											
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).											
If <b>Surface Format</b> is PLANAR*, this field must be set to TEXCOORDMODE_CLAMP.											
5:3	<table border="1"> <tr> <th colspan="2"><b>TCY Address Control Mode</b></th> </tr> <tr> <td>Format:</td> <td><b>Texture Coordinate Mode</b></td> </tr> <tr> <td colspan="2">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</td> </tr> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> <tr> <td colspan="2">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.</td> </tr> </table>	<b>TCY Address Control Mode</b>		Format:	<b>Texture Coordinate Mode</b>	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		<b>Programming Notes</b>		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.	
<b>TCY Address Control Mode</b>											
Format:	<b>Texture Coordinate Mode</b>										
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											
<b>Programming Notes</b>											
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.											
2:0	<table border="1"> <tr> <th colspan="2"><b>TCZ Address Control Mode</b></th> </tr> <tr> <td>Format:</td> <td><b>Texture Coordinate Mode</b></td> </tr> <tr> <td colspan="2">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</td> </tr> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> <tr> <td colspan="2">TCZ Address Control Mode Cannot use MIRROR_101 mode. MIRROR_101 mode only works for 2D surfaces.</td> </tr> </table>	<b>TCZ Address Control Mode</b>		Format:	<b>Texture Coordinate Mode</b>	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		<b>Programming Notes</b>		TCZ Address Control Mode Cannot use MIRROR_101 mode. MIRROR_101 mode only works for 2D surfaces.	
<b>TCZ Address Control Mode</b>											
Format:	<b>Texture Coordinate Mode</b>										
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											
<b>Programming Notes</b>											
TCZ Address Control Mode Cannot use MIRROR_101 mode. MIRROR_101 mode only works for 2D surfaces.											

## Sampler Message Header

SAMPLER_MSG_HEADER - Sampler Message Header		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
This defines the contents of the Message Header. Message Header the optional first 256-bits of any Sampler Message.		
DWord	Bit	Description
0	31:0	<b>Reserved</b>
		Format: MBZ
1	31:0	<b>Reserved</b>
		Format: MBZ
2	31	<b>Reserved</b>
		Format: MBZ
	30	<b>Reserved</b>
		Format: MBZ
	29:24	<b>Reserved</b>
	Format: MBZ	
23		<b>Pixel Null Mask Enable</b>
		Format: ENABLE
		<b>Pixel Null Mask Enable</b> Specifies whether the writeback message includes an extra phase indicating the pixel null mask. Refer to the <b>Writeback Message</b> section for details on format. This field must be disabled for ChromaKey and all SIMD32/64 messages.
		<b>Programming Notes</b> If this is set Shader channel select {Red Green Blue} must be set to zero for all corresponding missing color channels. Shader channel select Alpha must be set to one if the alpha channel is missing
22		<b>Return Filter Weights for Remaining On-Map Texels</b>
		Format: ENABLE
When set to 1: Sampler returns filter weights of remaining on-map texels. Filter weights are returned as a 32-bit float in the Red Channel. It can be used in conjunction with Return Filter Weights for Null Texels		
21		<b>Slot Group Select</b>
		<b>Slot Group Select</b> This field selects whether slots 7:0 or slots 15:8 are used for bypassed data. Bypassed data only includes the scale factors for CPS LOD Compensation. This field is ignored if <b>CPS Message LOD Compensation Enable</b> is disabled. For 8-pixel dispatches, SLOTGRP_0 must be selected on every message. For 16-pixel dispatches,



## SAMPLER\_MSG\_HEADER - Sampler Message Header

	<p>this field must be set correctly for each SIMD8* message based on which slots are currently being processed. For SIMD16* messages, SLOTGRP_0 must be selected.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>SLOTGRP_0 <b>[Default]</b></td> <td>Choose bypassed data for slots 7:0</td> </tr> <tr> <td>1h</td> <td>SLOTGRP_1</td> <td>Choose bypassed data for slots 15:8</td> </tr> </tbody> </table>	Value	Name	Description	0h	SLOTGRP_0 <b>[Default]</b>	Choose bypassed data for slots 7:0	1h	SLOTGRP_1	Choose bypassed data for slots 15:8	
Value	Name	Description									
0h	SLOTGRP_0 <b>[Default]</b>	Choose bypassed data for slots 7:0									
1h	SLOTGRP_1	Choose bypassed data for slots 15:8									
20	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
Access:	RO										
Format:	MBZ										
19:18	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ								
Format:	MBZ										
17:16	<p><b>Gather4 Source Channel Select</b>  <b>Gather4 Source Channel Select:</b> Selects the source channel to be sampled in the gather4* messages. Ignored for other message types.          For gather4*_c messages, this field must be set to 0 (Red channel).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>RED <b>[Default]</b></td> </tr> <tr> <td>1h</td> <td>GREEN</td> </tr> <tr> <td>2h</td> <td>BLUE</td> </tr> <tr> <td>3h</td> <td>ALPHA</td> </tr> </tbody> </table>	Value	Name	0h	RED <b>[Default]</b>	1h	GREEN	2h	BLUE	3h	ALPHA
Value	Name										
0h	RED <b>[Default]</b>										
1h	GREEN										
2h	BLUE										
3h	ALPHA										
15	<p><b>Alpha Write Channel Mask</b>          Enables the alpha channel to be written back to the originating thread.          0: Alpha channel is written back.          1: Alpha channel is not written back.          Restrictions for Channel Write Masks:</p> <ul style="list-style-type: none"> <li>• A message with all four channels masked is not allowed.</li> <li>• This field is ignored for the deinterlace message.</li> <li>• This field must be set to zero for all gather4* messages.</li> <li>• This field must be set to zero for sample_8x8 in VSA mode.</li> <li>• For Sample_8x8 messages, Alpha/Blue/Red channels should be always masked (set to 1) and only Green channel is enabled (set to 0).</li> </ul>										
14	<p><b>Blue Write Channel Mask</b>          Enables the blue channel to be written back to the originating thread.          See Alpha Channel Write Mask for usage restrictions.</p>										
13	<p><b>Green Write Channel Mask</b>          Enables the green channel to be written back to the originating thread.          See Alpha Channel Write Mask for usage restrictions.</p>										
12	<p><b>Red Write Channel Mask</b>          Enables the red channel to be written back to the originating thread.          See Alpha Channel Write Mask for usage restrictions.</p>										

## SAMPLER\_MSG\_HEADER - Sampler Message Header

	11:8	<b>U Offset</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S3</td> </tr> </table> <p><b>U Offset:</b>The u offset from the <code>_aoffimmi</code> modifier on the <code>sampleorld</code> instruction in DX10. Must be zero if the <b>Surface Type</b> is <code>SURFTYPE_CUBE</code> or <code>SURFTYPE_BUFFER</code>. Must be set to zero if <code>_aoffimmi</code> is not specified. Format is S3 2's complement.</p> <ul style="list-style-type: none"> <li>• This field is ignored for the <code>sample_unorm*</code>, <code>sample_8x8</code>, and <code>deinterlace</code> messages.</li> <li>• This field is ignored if the <code>offu</code> parameter is included in the <code>gather4*</code> messages.</li> </ul>	Format:	S3
Format:	S3				
	7:4	<b>V Offset</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S3</td> </tr> </table> <p><b>V Offset:</b>The v offset from the <code>_aoffimmi</code> modifier on the <code>sampleorld</code> instruction in DX10. Must be zero if the <b>Surface Type</b> is <code>SURFTYPE_CUBE</code> or <code>SURFTYPE_BUFFER</code>. Must be set to zero if <code>_aoffimmi</code> is not specified. Format is S3 2's complement.</p> <ul style="list-style-type: none"> <li>• This field is ignored for the <code>sample_unorm*</code>, <code>sample_8x8</code>, and <code>deinterlace</code> messages.</li> <li>• This field is ignored if the <code>offu</code> parameter is included in the <code>gather4*</code> messages.</li> </ul>	Format:	S3
Format:	S3				
	3:0	<b>R Offset</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">S3</td> </tr> </table> <p><b>R Offset:</b>The r offset from the <code>_aoffimmi</code> modifier on the <code>sampleorld</code> instruction in DX10. Must be zero if the <b>Surface Type</b> is <code>SURFTYPE_CUBE</code> or <code>SURFTYPE_BUFFER</code>. Must be set to zero if <code>_aoffimmi</code> is not specified. Format is S3 2's complement.</p> <p>This field is ignored for the <code>sample_unorm*</code>, <code>sample_8x8</code>, and <code>deinterlace</code> messages.</p> <p>Texel offsets can only be applied to messages with floating-point normalized coordinates or integer non-normalized coordinates.</p>	Format:	S3
Format:	S3				
3	31:4	<b>Sampler State Pointer</b>	<p><b>Sampler State Pointer:</b>Specifies the 16-byte aligned pointer to the sampler state table. This field is ignored for <code>ld</code> and <code>resin</code> message types. This pointer is relative to the <b>Dynamic State Base Address</b> or <b>Bindless Sampler State Base Address</b> depending on the setting of <b>Sampler State Base Address Select</b> field below.</p> <p>Format = <code>StateOffset[31:4]</code></p> <p>The Sampler State Pointer does not have to be defined by the Message Header (many messages do not require a message header). The Sampler State Pointer may be delivered from the Command Streamer without the need for a Message Header.</p>		
	3:1	<b>Reserved</b>	<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	<b>Sampler State Base Address Select</b>	<p><b>Sampler State Base Address Select:</b>Selects which base address is used for sampler state accesses.</p> <p>The Sampler State Base Address Select does not have to be defined by the Message Header (many messages do not require a message header). The Sampler State Base Address Select may be delivered from the Command Streamer without the need for a Message Header.</p>		

SAMPLER_MSG_HEADER - Sampler Message Header				
		Value	Name	Description
		0h	SAMP_DYNAMIC <b>[Default]</b>	Use Dynamic State Base Address
		1h	SAMP_BINDLESS	Use Bindless Sampler State Base Address
4	31:0	<b>Reserved</b>		
		Format:		MBZ
5	31:0	<b>Reserved</b>		
		Format:		MBZ
6	31:0	<b>Reserved</b>		
7	31:0	<b>Reserved</b>		



## SCALER\_COEFFICIENT\_FORMAT

SCALER_COEFFICIENT_FORMAT																			
Size (in bits):	16																		
Default Value:	0x00000000																		
Scaler coefficients are stored in sign-exponent-mantissa format. Two coefficients are stored in each dword, the table below show the data packing in each dword.																			
DWord	Bit	Description																	
0	15	<b>Sign</b>																	
		<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	<b>Reserved</b>																	
		Access:	RO																
		Format:	MBZ																
	13:12	<b>Exponent</b>																	
		All the tap coefficients use 2 bits of exponent.																	
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>2</td> <td>x2 or mantissa is b.bbbbbbbb</td> </tr> <tr> <td>01b</td> <td>1</td> <td>x1 or mantissa is 0.bbbbbbbb..</td> </tr> <tr> <td>10b</td> <td>0.5</td> <td>x0.5 or mantissa is 0.0bbbbbbb..</td> </tr> <tr> <td>11b</td> <td>0.25</td> <td>x0.25 or mantissa is 0.00bbbbbbb..</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Name	Description	00b	2	x2 or mantissa is b.bbbbbbbb	01b	1	x1 or mantissa is 0.bbbbbbbb..	10b	0.5	x0.5 or mantissa is 0.0bbbbbbb..	11b	0.25	x0.25 or mantissa is 0.00bbbbbbb..	Others	Reserved	Reserved
Value	Name	Description																	
00b	2	x2 or mantissa is b.bbbbbbbb																	
01b	1	x1 or mantissa is 0.bbbbbbbb..																	
10b	0.5	x0.5 or mantissa is 0.0bbbbbbb..																	
11b	0.25	x0.25 or mantissa is 0.00bbbbbbb..																	
Others	Reserved	Reserved																	
11:3	<b>Mantissa</b>																		
	All the tap coefficients use all 9 bits of mantissa.																		
2:0	<b>Reserved</b>																		
	Access:	RO																	
	Format:	MBZ																	

## SCISSOR\_RECT

SCISSOR_RECT				
Source:	RenderCS			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
<b>Restriction</b>				
When executed in the POCS command stream, this command programs the scissor state for the SFR stage of the POCS pipeline				
DWord	Bit	Description		
0	31:16	<b>Scissor Rectangle Y Min</b> Format: U16 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.		
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Value	Name
Value	Name			
[0,16383]				
15:0	15:0	<b>Scissor Rectangle X Min</b> Format: U16 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.		
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Value	Name
Value	Name			
[0,16383]				
1	31:16	<b>Scissor Rectangle Y Max</b> Format: U16 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.		
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Value	Name
Value	Name			
[0,16383]				
15:0	15:0	<b>Scissor Rectangle X Max</b> Format: U16 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.		
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0-16383</td> <td></td> </tr> </tbody> </table>	Value	Name
Value	Name			
0-16383				



## Scratch Hword Block Message Header

<b>MH_A32_HWB - Scratch Hword Block Message Header</b>		
Source:	EuSubFunctionDataPort0	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..2	95:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
3	31:0	<b>Per Thread Scratch Space</b>
		Format: <b>MHC_PTSS</b>
		Specifies amount of scratch space used by this thread, for Stateless bounds checking.
4	31:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
5	31:0	<b>Buffer Base Address</b>
		Format: <b>MHC_A32_BBA</b>
		Specifies the surface address offset page [31:10] for A32 stateless messages.
6..7	63:0	<b>Reserved</b>
		Access: RO
		Format: MBZ

## SF\_CLIP\_VIEWPORT

<b>SF_CLIP_VIEWPORT</b>		
Source:	RenderCS	
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
<b>Restriction</b>		
When executed in the POCS command stream, this command programs the viewport state for the CLR and SFR stage of the POCS pipeline.		
DWord	Bit	Description
0	31:0	<b>Viewport Matrix Element m00</b> Format: IEEE_FLOAT
1	31:0	<b>Viewport Matrix Element m11</b> Format: IEEE_FLOAT
2	31:0	<b>Viewport Matrix Element m22</b> Format: IEEE_FLOAT
3	31:0	<b>Viewport Matrix Element m30</b> Format: IEEE_FLOAT
4	31:0	<b>Viewport Matrix Element m31</b> Format: IEEE_FLOAT
5	31:0	<b>Viewport Matrix Element m32</b> Format: IEEE_FLOAT
6	31:0	<b>Reserved</b> Access: RO Format: MBZ
7	31:0	<b>Reserved</b> Access: RO Format: MBZ
8	31:0	<b>X Min Clip Guardband</b> Format: IEEE_FLOAT . 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.

## SF\_CLIP\_VIEWPORT

9	31:0	<p><b>X Max Clip Guardband</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">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	<p><b>Y Min Clip Guardband</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">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			
11	31:0	<p><b>Y Max Clip Guardband</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>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.</p>	Format:	IEEE_FLOAT
Format:	IEEE_FLOAT			
12	31:0	<p><b>X Min ViewPort</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>This 32-bit float represents the Viewport.XMin.</p> <p>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.</p>	Format:	IEEE_FLOAT
Format:	IEEE_FLOAT			
13	31:0	<p><b>X Max ViewPort</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>This 32-bit float represents the Viewport.XMax.</p> <p>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.</p>	Format:	IEEE_FLOAT
Format:	IEEE_FLOAT			
14	31:0	<p><b>Y Min ViewPort</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>This 32-bit float represents the Viewport.YMin.</p> <p>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.</p>	Format:	IEEE_FLOAT
Format:	IEEE_FLOAT			
15	31:0	<p><b>Y Max ViewPort</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>IEEE_FLOAT</td> </tr> </table> <p>This 32-bit float represents the Viewport.Ymax.</p> <p>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.</p>	Format:	IEEE_FLOAT
Format:	IEEE_FLOAT			



## SF\_OUTPUT\_ATTRIBUTE\_DETAIL

SF_OUTPUT_ATTRIBUTE_DETAIL				
Source:	RenderCS			
Size (in bits):	16			
Default Value:	0x00000000			
DWord	Bit	Description		
0	15	<p><b>Component Override W</b></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><b>Component Override Z</b></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><b>Component Override Y</b></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><b>Component Override X</b></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><b>Swizzle Control Mode</b></p> <table border="1"> <tr> <td>Format:</td> <td>U1</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"> <li>• Component Override X/Y/Z/W</li> <li>• Constant Source</li> <li>• Swizzle Select</li> <li>• Source Attribute</li> <li>• WrapShortest Enables</li> </ul> <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	
Format:	U1			

## SF\_OUTPUT\_ATTRIBUTE\_DETAIL

10:9	<b>Constant Source</b>		
		Format:	U2
This state selects a constant vector which can be used to override individual components of this Attribute			
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
		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	<b>Reserved</b>		
		Access:	RO
		Format:	MBZ
7:6	<b>Swizzle Select</b>		
		Format:	U2
This state, along with Source Attribute, specifies the source for this output Attribute.			
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
		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].
		2h	INPUTATTR_W
			This attribute is sourced from AttrInputReg[SourceAttribute]. The W component is copied to the X component.
		3h	INPUTATTR_FACING_W
			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]. The W component is copied to the X component.
5	<b>Reserved</b>		
		Access:	RO
		Format:	MBZ
4:0	<b>Source Attribute</b>		
		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		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
ExistsIf = AVS && (Function_mode = 0)		
DWord	Bit	Description
0	31:24	<b>ZeroYFilterCoefficient1</b>
		Format: S1.6 Range: [-2, +2)
	23:16	<b>ZeroXFilterCoefficient1</b>
		Format: S1.6 Range: [-2, +2)
15:8	<b>ZeroYFilterCoefficient0</b>	
	Format: S1.6 Range: [-2, +2)	
7:0	<b>ZeroXFilterCoefficient0</b>	
		Format: S1.6 Range: [-2, +2)
1	31:24	<b>ZeroYFilterCoefficient3</b>
		Format: S1.6 Range: [-2, +2)
	23:16	<b>ZeroXFilterCoefficient3</b>
		Format: S1.6 Range: [-2, +2)
15:8	<b>ZeroYFilterCoefficient2</b>	
	Format: S1.6 Range: [-2, +2)	
7:0	<b>ZeroXFilterCoefficient2</b>	
		Format: S1.6 Range: [-2, +2)

<b>SFC_8x8_AVS_COEFFICIENTS</b>				
2	31:24	<b>ZeroYFilterCoefficient5</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6
		S1.6		
	23:16	<b>ZeroXFilterCoefficient5</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6
		S1.6		
15:8	<b>ZeroYFilterCoefficient4</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6	
	S1.6			
7:0	<b>ZeroXFilterCoefficient4</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6	
	S1.6			
3	31:24	<b>ZeroYFilterCoefficient7</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6
		S1.6		
	23:16	<b>ZeroXFilterCoefficient7</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6
		S1.6		
15:8	<b>ZeroYFilterCoefficient6</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6	
	S1.6			
7:0	<b>ZeroXFilterCoefficient6</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6	
	S1.6			
4	31:24	<b>OneXFilterCoefficient3</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-2.0, +2.0)		S1.6
		S1.6		
23:16	<b>OneXFilterCoefficient2</b> Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">S1.6</td></tr></table> <b>Range:</b> [-1.0, +1.0)		S1.6	
	S1.6			

<b>SFC_8x8_AVS_COEFFICIENTS</b>		
	15:0	<b>Reserved</b> Access: RO Format: MBZ
5	31:16	<b>Reserved</b> Access: RO Format: MBZ
	15:8	<b>OneXFilterCoefficient5</b> Format: S1.6 <b>Range:</b> [-1.0, +1.0)
	7:0	<b>OneXFilterCoefficient4</b> Format: S1.6 <b>Range:</b> [-2.0, +2.0)
6	31:24	<b>OneYFilterCoefficient3</b> Format: S1.6 <b>Range:</b> [-2.0, +2.0)
	23:16	<b>OneYFilterCoefficient2</b> Format: S1.6 <b>Range:</b> [-1.0, +1.0)
	15:0	<b>Reserved</b> Access: RO Format: MBZ
7	31:16	<b>Reserved</b> Access: RO Format: MBZ
	15:8	<b>OneYFilterCoefficient5</b> Format: S1.6 <b>Range:</b> [-1.0, +1.0)
	7:0	<b>OneYFilterCoefficient4</b> Format: S1.6 <b>Range:</b> [-2.0, +2.0)

## SFC\_AVS\_CHROMA\_COEFF\_TABLE\_BODY

SFC_AVS_CHROMA_COEFF_TABLE_BODY		
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:56	<b>Table 1Y Filter Coefficient[[n],5]</b>
		Format: <span style="float: right;">S1.6</span>
		<b>Range:</b> [-2, +2] Chroma table for Y-direction.
		<b>Programming Notes</b> Filter tap index3 in U/V 4-tap filtering
	55:48	<b>Table 1X Filter Coefficient[[n],5]</b>
		Format: <span style="float: right;">S1.6</span>
		<b>Range:</b> [-2, +2] Chroma table for X-direction.
		<b>Programming Notes</b> Filter tap index3 in U/V 4-tap filtering
	47:40	<b>Table 1Y Filter Coefficient[[n],4]</b>
		Format: <span style="float: right;">S1.6</span>
		<b>Range:</b> [-2, +2] Chroma table for Y-direction.
		<b>Programming Notes</b> Filter tap index 2 in U/V 4-tap filtering
	39:32	<b>Table 1X Filter Coefficient[[n],4]</b>
		Format: <span style="float: right;">S1.6</span>
		<b>Range:</b> [-2, +2] Chroma table for X-direction.
		<b>Programming Notes</b> Filter tap index 2 in U/V 4-tap filtering

## SFC\_AVS\_CHROMA\_COEFF\_TABLE\_BODY

	31:24	<b>Table 1Y Filter Coefficient[[n],3]</b> Format: <span style="float: right;">S1.6</span>  <b>Range:</b> [-2, +2) Chroma table for Y-direction.  <div style="text-align: center;"><b>Programming Notes</b></div> Filter tap index1 in U/V 4-tap filtering
	23:16	<b>Table 1X Filter Coefficient[[n],3]</b> Format: <span style="float: right;">S1.6</span>  <b>Range:</b> [-2, +2) Chroma table for X-direction.  <div style="text-align: center;"><b>Programming Notes</b></div> Filter tap index1 in U/V 4-tap filtering
	15:8	<b>Table 1Y Filter Coefficient[[n],2]</b> Format: <span style="float: right;">S1.6</span>  <b>Range:</b> [-2, +2) Chroma table for Y-direction.  <div style="text-align: center;"><b>Programming Notes</b></div> Filter tap index0 in U/V 4-tap filtering
	7:0	<b>Table 1X Filter Coefficient[[n],2]</b> Format: <span style="float: right;">S1.6</span>  <b>Range:</b> [-2, +2) Chroma table for X-direction.  <div style="text-align: center;"><b>Programming Notes</b></div> Filter tap index0 in U/V 4-tap filtering

## SFC\_AVS\_LUMA\_COEFF\_TABLE\_BODY

SFC_AVS_LUMA_COEFF_TABLE_BODY		
Size (in bits):	128	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..3	127:120	<b>Table 0Y Filter Coefficient[[n],7]</b>
		Format: <span style="float: right;">S1.6</span>
		<b>Range:</b> [-2, +2] Luma table for Y-direction.
		<b>Programming Notes</b> For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.
	119:112	<b>Table 0X Filter Coefficient[[n],7]</b>
		Format: <span style="float: right;">S1.6</span>
		<b>Range:</b> [-2, +2] Luma table for X-direction.
		<b>Programming Notes</b> For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.
	111:104	<b>Table 0Y Filter Coefficient[[n],6]</b>
		Format: <span style="float: right;">S1.6</span>
		<b>Range:</b> [-2, +2] Luma table for Y-direction.
		<b>Programming Notes</b> For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.
	103:96	<b>Table 0X Filter Coefficient[[n],6]</b>
		Format: <span style="float: right;">S1.6</span>
		<b>Range:</b> [-2, +2] Luma table for X-direction.
		<b>Programming Notes</b> For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.



## SFC\_AVS\_LUMA\_COEFF\_TABLE\_BODY

	95:88	<b>Table 0Y Filter Coefficient[[n],5]</b>	
		Format:	S1.6
		<b>Range:</b> [-2, +2)	
		Luma table for Y-direction.	
		<b>Programming Notes</b>	
		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
	87:80	<b>Table 0X Filter Coefficient[[n],5]</b>	
		Format:	S1.6
		<b>Range:</b> [-2, +2)	
		Luma table for X-direction.	
		<b>Programming Notes</b>	
		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
	79:72	<b>Table 0Y Filter Coefficient[[n],4]</b>	
		Format:	S1.6
		<b>Range:</b> [-2, +2)	
		Luma table for Y-direction.	
		<b>Programming Notes</b>	
		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
	71:64	<b>Table 0X Filter Coefficient[[n],4]</b>	
		Format:	S1.6
<b>Range:</b> [-2, +2)			
Luma table for X-direction.			
<b>Programming Notes</b>			
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.			
63:56	<b>Table 0Y Filter Coefficient[[n],3]</b>		
	Format:	S1.6	
	<b>Range:</b> [-2, +2)		
	Luma table for Y-direction.		

<b>SFC_AVS_LUMA_COEFF_TABLE_BODY</b>													
	<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">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.									
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
55:48	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: left;"><b>Table 0X Filter Coefficient[[n],3]</b></th> </tr> <tr> <td style="width: 60%;">Format:</td> <td>S1.6</td> </tr> <tr> <td colspan="2"><b>Range:</b> [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for X-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	<b>Table 0X Filter Coefficient[[n],3]</b>		Format:	S1.6	<b>Range:</b> [-2, +2)		Luma table for X-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
<b>Table 0X Filter Coefficient[[n],3]</b>													
Format:	S1.6												
<b>Range:</b> [-2, +2)													
Luma table for X-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
47:40	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: left;"><b>Table 0Y Filter Coefficient[[n],2]</b></th> </tr> <tr> <td style="width: 60%;">Format:</td> <td>S1.6</td> </tr> <tr> <td colspan="2"><b>Range:</b> [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for Y-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	<b>Table 0Y Filter Coefficient[[n],2]</b>		Format:	S1.6	<b>Range:</b> [-2, +2)		Luma table for Y-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
<b>Table 0Y Filter Coefficient[[n],2]</b>													
Format:	S1.6												
<b>Range:</b> [-2, +2)													
Luma table for Y-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
39:32	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: left;"><b>Table 0X Filter Coefficient[[n],2]</b></th> </tr> <tr> <td style="width: 60%;">Format:</td> <td>S1.6</td> </tr> <tr> <td colspan="2"><b>Range:</b> [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for X-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	<b>Table 0X Filter Coefficient[[n],2]</b>		Format:	S1.6	<b>Range:</b> [-2, +2)		Luma table for X-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
<b>Table 0X Filter Coefficient[[n],2]</b>													
Format:	S1.6												
<b>Range:</b> [-2, +2)													
Luma table for X-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													
31:24	<table border="1" style="width: 100%;"> <tr> <th colspan="2" style="text-align: left;"><b>Table 0Y Filter Coefficient[[n],1]</b></th> </tr> <tr> <td style="width: 60%;">Format:</td> <td>S1.6</td> </tr> <tr> <td colspan="2"><b>Range:</b> [-2, +2)</td> </tr> <tr> <td colspan="2">Luma table for Y-direction.</td> </tr> <tr> <th colspan="2" style="text-align: center; background-color: #e6f2ff;">Programming Notes</th> </tr> <tr> <td colspan="2">For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.</td> </tr> </table>	<b>Table 0Y Filter Coefficient[[n],1]</b>		Format:	S1.6	<b>Range:</b> [-2, +2)		Luma table for Y-direction.		Programming Notes		For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.	
<b>Table 0Y Filter Coefficient[[n],1]</b>													
Format:	S1.6												
<b>Range:</b> [-2, +2)													
Luma table for Y-direction.													
Programming Notes													
For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.													

<b>SFC_AVS_LUMA_COEFF_TABLE_BODY</b>		
	23:16	<b>Table 0X Filter Coefficient[[n],1]</b> Format: <span style="float: right;">S1.6</span>  <b>Range:</b> [-2, +2) Luma table for X-direction.  <div style="text-align: center; background-color: #e6f2ff; padding: 2px;"><b>Programming Notes</b></div> For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.
	15:8	<b>Table 0Y Filter Coefficient[[n],0]</b> Format: <span style="float: right;">S1.6</span>  <b>Range:</b> [-2, +2) Luma table for Y-direction.  <div style="text-align: center; background-color: #e6f2ff; padding: 2px;"><b>Programming Notes</b></div> For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.
	7:0	<b>Table 0X Filter Coefficient[[n],0]</b> Format: <span style="float: right;">S1.6</span>  <b>Range:</b> [-2, +2) Luma table for X-direction.  <div style="text-align: center; background-color: #e6f2ff; padding: 2px;"><b>Programming Notes</b></div> For 5x5 filter mode, table 0, 6 and 7 must be programmed to zero.

## SFC\_AVS\_STATE\_BODY

SFC_AVS_STATE_BODY											
Size (in bits):	96										
Default Value:	0x00000000, 0x00000000, 0x00000000										
DWord	Bit	Description									
0	31:24	<b>Sharpness Level</b>									
		Format: U8									
		When adaptive scaling is off, determines the balance between sharp and smooth scalers.									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></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		Contribute 1 from the smooth scalar	255		Contribute 1 from the sharp scalar
		Value	Name	Description							
	0		Contribute 1 from the smooth scalar								
	255		Contribute 1 from the sharp scalar								
	23:7	<b>Reserved</b>									
	Access: RO										
	Format: MBZ										
6:4	<b>Transition Area with 4 Pixels</b>										
	Format: U3										
		Used in adaptive filtering to specify the width of the transition area for the 4 pixel calculation.									
3	<b>Reserved</b>										
	Access: RO										
	Format: MBZ										
2:0	<b>Transition Area with 8 Pixels</b>										
	Format: U3										
	Used in adaptive filtering to specify the width of the transition area for the 8 pixel calculation.										
1	31:24	<b>Reserved</b>									
		Access: RO									
		Format: MBZ									
	23:16	<b>Max Derivative 4 Pixels</b>									
		Format: U8									
			Used in adaptive filtering to specify the lower boundary of the smooth 4 pixel area.								
	15:8	<b>Reserved</b>									
		Access: RO									
		Format: MBZ									

<b>SFC_AVS_STATE_BODY</b>																						
	7:0	<b>MAX Derivative Point 8</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">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																				
2	31:13	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																
		Access:	RO																			
	Format:	MBZ																				
	12	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																
Access:		RO																				
Format:	MBZ																					
11:8	<b>Input Horizontal Siting Value - Specifies the horizontal siting of the input</b>																					
	<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 30%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr><td>0000b</td><td>0(fraction in integer)</td></tr> <tr><td>0001b</td><td>1/8</td></tr> <tr><td>0010b</td><td>2/8</td></tr> <tr><td>0011b</td><td>3/8</td></tr> <tr><td>0100b</td><td>4/8</td></tr> <tr><td>0101b</td><td>5/8</td></tr> <tr><td>0110b</td><td>6/8</td></tr> <tr><td>0111b</td><td>7/8</td></tr> <tr><td>1000b</td><td>8/8</td></tr> </tbody> </table>		Value	Name	0000b	0(fraction in integer)	0001b	1/8	0010b	2/8	0011b	3/8	0100b	4/8	0101b	5/8	0110b	6/8	0111b	7/8	1000b	8/8
	Value	Name																				
	0000b	0(fraction in integer)																				
	0001b	1/8																				
	0010b	2/8																				
	0011b	3/8																				
	0100b	4/8																				
	0101b	5/8																				
	0110b	6/8																				
0111b	7/8																					
1000b	8/8																					
<b>Programming Notes</b>																						
For 444 format, horizontal chroma siting should be programmed to zero.																						
7:4	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ																
	Access:	RO																				
Format:	MBZ																					
3:0	<b>Input Vertical Siting - Specifies the vertical siting of the input</b>																					
	<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="width: 50%;">Value</th> <th>Name</th> </tr> </thead> <tbody> <tr><td>000b</td><td>0</td></tr> <tr><td>0001b</td><td>1/8</td></tr> <tr><td>0010b</td><td>2/8</td></tr> <tr><td>0011b</td><td>3/8</td></tr> <tr><td>0100b</td><td>4/8</td></tr> <tr><td>0101b</td><td>5/8</td></tr> <tr><td>0110b</td><td>6/8</td></tr> </tbody> </table>		Value	Name	000b	0	0001b	1/8	0010b	2/8	0011b	3/8	0100b	4/8	0101b	5/8	0110b	6/8				
	Value	Name																				
	000b	0																				
	0001b	1/8																				
	0010b	2/8																				
	0011b	3/8																				
	0100b	4/8																				
0101b	5/8																					
0110b	6/8																					

<b>SFC_AVS_STATE_BODY</b>					
	<table border="1"><tr><td>0111b</td><td>7/8</td></tr><tr><td>1000b</td><td>8/8</td></tr></table>	0111b	7/8	1000b	8/8
0111b	7/8				
1000b	8/8				
	<table border="1"><tr><td colspan="2"><b>Programming Notes</b></td></tr><tr><td colspan="2">For 444 and 422 format, vertical chroma siting should be programmed to zero.</td></tr></table>	<b>Programming Notes</b>		For 444 and 422 format, vertical chroma siting should be programmed to zero.	
<b>Programming Notes</b>					
For 444 and 422 format, vertical chroma siting should be programmed to zero.					

## SFC\_FRAME\_START\_BODY

SFC_FRAME_START_BODY		
Size (in bits):		32
Default Value:		0x00000000
DWord	Bit	Description
0	31:0	<b>Reserved</b>
		Access: RO
		Format: MBZ

## SFC\_HDR\_STATE

<b>SFC_HDR_STATE</b>																	
Size (in bits):		96															
Default Value:		0x00000000, 0x00000000, 0x00000000															
DWord	Bit	Description															
0..1	63:48	<b>Reserved</b>															
		Access: RO															
		Format: MBZ															
	47:12	<b>Address</b>															
		Format: GraphicsAddress[47:12] Specifies the graphics base address used to fetch SFC_EOTF_OETF_STATE surfacetable into SFC.															
	11:0	<b>Reserved</b>															
		Access: RO															
		Format: MBZ															
2	31:15	<b>Reserved</b>															
		Access: RO															
	14:13	<b>Surface Tiled Mode</b>															
		Format: U2 <b>For Media Surfaces:</b> 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>0</td> <td>TRMODE_NONE</td> <td>No tiled resources</td> </tr> <tr> <td>1</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Value	Name	Description	0	TRMODE_NONE	No tiled resources	1	TRMODE_TILEYF	4KB tiled resources	2	TRMODE_TILEYS	64KB tiled resources	3	Reserved	
Value	Name	Description															
0	TRMODE_NONE	No tiled resources															
1	TRMODE_TILEYF	4KB tiled resources															
2	TRMODE_TILEYS	64KB tiled resources															
3	Reserved																
	12	<b>Reserved</b>															
		Access: RO															
	11	<b>Scratch Buffer Cache Select</b>															
		Default Value: 0 Disable															
		Format: U1															
		<b>Programming Notes</b>															
		This must be set to 0															



<b>SFC_HDR_STATE</b>	
10	<b>Compression Type</b>
	Default Value: 0 Disable
	Format: boolean
	This field is applicable only when Memory compression is enabled. As memory compression is not supported on this surface, it must be 0.
9	<b>Memory Compression Enable</b>
	Default Value: 0 Disable
	Format: Enable
	<b>Programming Notes</b>
Memory compression is not supported for this surface Must be 0.	
8:7	<b>Arbitration Priority Control</b>
	Format: <b>HEVC_ARBITRATION_PRIORITY</b>
6:1	<b>Index to Memory Object Control State (MOCS) Tables</b>
	Format: U6
The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers. The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
0	<b>Reserved</b>



## SFC\_HIST\_SURF - SFC\_HISTOGRAM\_SURFACE

0x00000000, 0x00000000, 0x00000000, 0x00000000

SFC would output histogram to a surface with 256-bins when histogram streamout is enabled in SFC\_STATE. This represent the write out surface structure.

DWord	Bit	Description		
0..3	127:96	<b>HistogramBinLumaY[3]</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table>	Format:	U32
	Format:	U32		
	95:64	<b>HistogramBinLumaY[2]</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table>	Format:	U32
	Format:	U32		
63:32	<b>HistogramBinLumaY[1]</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table>	Format:	U32	
Format:	U32			
31:0	<b>HistogramBinLumaY[0]</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U32</td> </tr> </table>	Format:	U32	
Format:	U32			
4..255	31:0	<b>HistogramBinLumaY[4..255]</b>		

## SFC\_IEF\_STATE\_BODY

<b>SFC_IEF_STATE_BODY</b>		
Size (in bits):	736	
Default Value:	0x0294806C, 0x39CFD1FF, 0x039F0000, 0x9A6E4000, 0x00601180, 0xFFFE2F2E, 0x00000000, 0xD82E0000, 0x8285ECEC, 0x00008282, 0x00000000, 0x02117000, 0xA38FEC96, 0x00008CC8, 0x00000000, 0x00002000, 0x00000000, 0x00000400, 0x00000000, 0x00000400, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:28	<b>Reserved</b>
		Access: RO Format: MBZ
	27:23	<b>R3c Coefficient</b>
		Default Value: 5
		Format: U0.5 IEF smoothing coefficient, <i>see IEF map</i> .
	22:18	<b>R3x Coefficient</b>
		Default Value: 5
		Format: U0.5 IEF smoothing coefficient, <i>see IEF map</i> .
	17:12	<b>Strong Edge Threshold</b>
		Default Value: 8
		Format: U6 If EM > Strong Edge Threshold the basic VSA detects a strong edge.
	11:6	<b>Weak Edge Threshold</b>
		Default Value: 1
		Format: U6 If Strong Edge Threshold > EM > Weak Edge Threshold the basic VSA detects a weak edge.
	5:0	<b>Gain Factor</b>
		Default Value: 44
		Format: U6 User control sharpening strength.

<b>SFC_IEF_STATE_BODY</b>		
1	31:27	<b>R5c Coefficient</b>
		Default Value: 7
		Format: U0.5 IEF smoothing coefficient, <i>see IEF map.</i>
	26:22	<b>R5cx Coefficient</b>
		Default Value: 7
		Format: U0.5 IEF smoothing coefficient, <i>see IEF map.</i>
	21:17	<b>R5x Coefficient</b>
Default Value: 7		
Format: U0.5 IEF smoothing coefficient, <i>see IEF map.</i>		
16:14	<b>Strong Edge Weight</b>	
	Default Value: 7	
	Format: U3 Sharpening strength when a <u>STRONG</u> edge is found in basic VSA.	
13:11	<b>Regular Weight</b>	
	Default Value: 2	
	Format: U3 Sharpening strength when a <u>WEAK</u> edge is found in basic VSA.	
10:8	<b>Non Edge Weight</b>	
	Default Value: 1	
	Format: U3 . Sharpening strength when <u>NO EDGE</u> is found in basic VSA.	
7:0	<b>Global Noise Estimation</b>	
	Default Value: 255	
	Format: U8 Global noise estimation of previous frame.	
2	31:28	<b>Reserved</b>
		Access: RO
		Format: MBZ

<b>SFC_IEF_STATE_BODY</b>			
	27:22	<b>Hue_Max</b>	
		Default Value: 14	
		Format: U6	
	Rectangle half width.		
	21:16	<b>Sat_Max</b>	
		Default Value: 31	
		Format: U6	
	Rectangle half length.		
	15:8	<b>STD Cos(alpha)</b>	
		Format: S0.7	
	<b>Default Value = 79/128</b>		
	7:0	<b>STD Sin(alpha)</b>	
Format: S0.7			
<b>Default Value = 101/128</b>			
3	31:24	<b>V_Mid</b>	
		Default Value: 154	
		Format: U8	
	Rectangle middle-point V coordinate.		
	23:16	<b>U_Mid</b>	
		Default Value: 110	
		Format: U8	
	Rectangle middle-point U coordinate.		
	15	<b>VY_STD_Enable</b>	
		Format: Enable	
	Enables STD in the VY subspace.		
	14:12	<b>Diamond Margin</b>	
Default Value: 4			
Format: U3			
11	<b>Reserved</b>		
	Access: RO		
	Format: MBZ		

<b>SFC_IEF_STATE_BODY</b>						
	10:0	<b>S3U</b> Format: S2.8 Slope 3 of the upper part of the detection PWLF. <b>Deafult:</b> 0/256				
		<b>Skin Detail Factor</b> Format: U1 This flag bit is in operation only when one of the following conditions exists: <ul style="list-style-type: none"> <li>when the control bit <b>SkinToneTunedIEF_Enable</b> is on.</li> <li>When <b>SkinDetailFactor</b> is equal to 0, sign(<b>SkinDetailFactor</b>) is equal to +1, and the content of the detected skin tone area is detail revealed. When <b>SkinDetailFactor</b> is equal to 1, sign(<b>SkinDetailFactor</b>) is equal to -1, and the content of the detected skin tone area is not detail revealed.</li> </ul> <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>Detail Revealed <b>[Default]</b></td> </tr> <tr> <td style="text-align: center;">1</td> <td>Not Detail Revealed</td> </tr> </tbody> </table>	Value	Name	0	Detail Revealed <b>[Default]</b>
Value	Name					
0	Detail Revealed <b>[Default]</b>					
1	Not Detail Revealed					
	30:24	<b>Diamond_du</b> Default Value: 0 Format: S6 Rhombus center shift in the sat-direction, relative to the rectangle center.				
		<b>HS_margin</b> Default Value: 3 Format: U3 Defines rectangle margin.				
	20:13	<b>Diamond_alpha</b> Format: U2.6 $1 / \tan()$ <b>Deafult:</b> 100/64				
		<b>Diamond_Th</b> Default Value: 35 Format: U6 Half length of the rhombus axis in the sat-direction.				

<b>SFC_IEF_STATE_BODY</b>						
	6:0	<b>Diamond_dv</b>				
		<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>S6</td> </tr> </table> <p>Rhombus center shift in the hue-direction, relative to the rectangle center.</p>	Default Value:	0	Format:	S6
Default Value:	0					
Format:	S6					
5	31:24	<b>Y_point_4</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">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				
23:16	<b>Y_point_3</b>					
<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">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					
15:8	<b>Y_point_2</b>					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">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					
7:0	<b>Y_point_1</b>					
<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">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					
6	31:16	<b>Reserved</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Access:</td> <td style="width: 50%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
15:0	<b>INV_Margin_VYL</b>					
<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U0.16</td> </tr> </table> <p>1 / Margin_VYL</p> <p><b>Default:</b> 3300/65536</p>		Format:	U0.16			
Format:	U0.16					
7	31:24	<b>P1L</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">216</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Y Point 1 of the lower part of the detection PWLF.</p>		Default Value:	216	Format:	U8
	Default Value:	216				
Format:	U8					



<b>SFC_IEF_STATE_BODY</b>						
	23:16	<b>P0L</b> <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> <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	<b>INV_Margin_VYU</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U0.16</td> </tr> </table> <p>1 / Margin_VYL</p> <p><b>Default:</b> 1600/65536</p>	Format:	U0.16		
Format:	U0.16					
8	31:24	<b>B1L</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">130</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>B0L</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">133</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>P3L</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">236</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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	<b>P2L</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">236</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">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					
9	31:27	<b>Y_Slope_2</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: center;">U2.3</td> </tr> </table> <p>Slope between points Y3 and Y4.</p> <p><b>Default:</b> 31/8</p>	Format:	U2.3		
Format:	U2.3					

<b>SFC_IEF_STATE_BODY</b>		
	26:16	<b>S0L</b>
		Format:   S2.8
		Slope 0 of the lower part of the detection PWLF. <b>Default:</b> -5/256
	15:8	<b>B3L</b>
		Default Value:   130
		Format:   U8 V Bias 3 of the lower part of the detection PWLF.
	7:0	<b>B2L</b>
		Default Value:   130
		Format:   U8 V Bias 2 of the lower part of the detection PWLF.
10	31:22	<b>Reserved</b>
		Access:   RO Format:   MBZ
	21:11	<b>S2L</b>
		Format:   S2.8
		<b>Default:</b> 0/256 Slope 2 of the lower part of the detection PWLF.
	10:0	<b>S1L</b>
Format:   S2.8		
<b>Default:</b> 0/256 Slope 1 of the lower part of the detection PWLF.		
11	31:27	<b>Y_Slope1</b>
		Format:   U2.3
		Slope between points Y1 and Y2. <b>Default:</b> 31/8
	26:19	<b>P1U</b>
		Default Value:   66 Format:   U8 Y Point 1 of the upper part of the detection PWLF.

<b>SFC_IEF_STATE_BODY</b>						
	18:11	<b>P0U</b> <table border="1"> <tr> <td>Default Value:</td> <td>46</td> </tr> <tr> <td>Format:</td> <td>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	<b>S3L</b> <table border="1"> <tr> <td>Format:</td> <td>S2.8</td> </tr> </table> <p>Slope 3 of the lower part of the detection PWLF.</p> <p><b>Default:</b> 0/256</p>	Format:	S2.8		
Format:	S2.8					
12	31:24	<b>B1U</b> <table border="1"> <tr> <td>Default Value:</td> <td>163</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 1 of the upper part of the detection PWLF.</p>	Default Value:	163	Format:	U8
	Default Value:	163				
	Format:	U8				
	23:16	<b>B0U</b> <table border="1"> <tr> <td>Default Value:</td> <td>143</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 0 of the upper part of the detection PWLF.</p>	Default Value:	143	Format:	U8
Default Value:	143					
Format:	U8					
15:8	<b>P3U</b> <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 upper part of the detection PWLF.</p>	Default Value:	236	Format:	U8	
Default Value:	236					
Format:	U8					
	7:0	<b>P2U</b> <table border="1"> <tr> <td>Default Value:</td> <td>150</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Y Point 2 of the upper part of the detection PWLF.</p>	Default Value:	150	Format:	U8
Default Value:	150					
Format:	U8					
13	31:27	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
	26:16	<b>S0U</b> <table border="1"> <tr> <td>Format:</td> <td>S2.8</td> </tr> </table> <p>Slope 0 of the upper part of the detection PWLF.</p> <p><b>Default:</b> 256/256</p>	Format:	S2.8		
Format:	S2.8					

<b>SFC_IEF_STATE_BODY</b>						
	15:8	<b>B3U</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">140</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> V Bias 3 of the upper part of the detection PWLF.	Default Value:	140	Format:	U8
	Default Value:	140				
Format:	U8					
7:0	<b>B2U</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">200</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> V Bias 2 of the upper part of the detection PWLF.	Default Value:	200	Format:	U8	
Default Value:	200					
Format:	U8					
14	31:22	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: right;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
21:11	<b>S2U</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">S2.8</td> </tr> </table> <b>Default:</b> -179/256 Slope 2 of the upper part of the detection PWLF.	Format:	S2.8			
Format:	S2.8					
10:0	<b>S1U</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td style="text-align: right;">S2.8</td> </tr> </table> <b>Default:</b> 113/256 Slope 1 of the upper part of the detection PWLF.	Format:	S2.8			
Format:	S2.8					
15	31:29	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: right;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
28:16	<b>C1</b> <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;">S2.10</td> </tr> </table> Transform coefficient	Default Value:	0	Format:	S2.10	
Default Value:	0					
Format:	S2.10					
15:3	<b>C0</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">1024</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">S2.10</td> </tr> </table> Transform coefficient	Default Value:	1024	Format:	S2.10	
Default Value:	1024					
Format:	S2.10					

<b>SFC_IEF_STATE_BODY</b>		
	2	<b>Reserved</b>
		Access: RO
		Format: MBZ
	1	<b>YUV Channel Swap</b>
	0	<b>Transform Enable</b>
16	31:26	<b>Reserved</b>
		Access: RO
		Format: MBZ
	25:13	<b>C3</b>
		Default Value: 0
		Format: S2.10
	Transform coefficient	
	12:0	<b>C2</b>
		Default Value: 0
		Format: S2.10
	Transform coefficient	
17	31:26	<b>Reserved</b>
		Access: RO
		Format: MBZ
	25:13	<b>C5</b>
		Default Value: 0
		Format: S2.10
	Transform coefficient	
	12:0	<b>C4</b>
		Default Value: 1024
		Format: S2.10
	Transform coefficient	
18	31:26	<b>Reserved</b>
		Access: RO
		Format: MBZ
	25:13	<b>C7</b>
		Default Value: 0
	Format: S2.10	
	Transform coefficient	

<b>SFC_IEF_STATE_BODY</b>		
	12:0	<b>C6</b>
		Default Value: 0
		Format: S2.10 Transform coefficient
19	31:13	<b>Reserved</b>
		Access: RO Format: MBZ
	12:0	<b>C8</b>
		Default Value: 1024
		Format: S2.10 Transform coefficient
	20	31:22
Access: RO Format: MBZ		
21:11		<b>Offset out 1</b>
		Default Value: 0
		Format: S2.8 Offset out for Y/R.
10:0		<b>Offset in 1</b>
	Default Value: 0	
	Format: S2.8 Offset in for Y/R.	
21	31:22	<b>Reserved</b>
		Access: RO Format: MBZ
	21:11	<b>Offset out 2</b>
		Default Value: 0
		Format: S2.8 Offset out for U/G.
	10:0	<b>Offset in 2</b>
Default Value: 0		
Format: S2.8 Offset in for U/G.		

<b>SFC_IEF_STATE_BODY</b>						
22	31:22	<b>Reserved</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	21:11	<b>Offset out 3</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S2.8</td> </tr> </table>	Default Value:	0	Format:	S2.8
		Default Value:	0			
	Format:	S2.8				
	Offset out for V/B.					
10:0	<b>Offset in 3</b>					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S2.8</td> </tr> </table>	Default Value:	0	Format:	S2.8	
	Default Value:	0				
Format:	S2.8					
Offset in for V/B.						

## SFC\_LOCK\_BODY

<b>SFC_LOCK_BODY</b>																							
Size (in bits):		32																					
Default Value:		0x00000000																					
DWord	Bit	Description																					
0	31:3	<b>Reserved</b>																					
		Access:	RO																				
		Format:	MBZ																				
	2	<b>Reserved</b>																					
		Access:	RO																				
		Format:	MBZ																				
	1	<b>Pre-Scaled Output Surface Output Enable</b>																					
		<b>VD - Reconstructed Pixel Output Enable</b>																					
		For VD Mode, this field specifies the enabling of writing out the display reconstructed pixel to memory. It could be pre or post- ILDB filter pixel output based on the pre- and post- filter setting in the AVC state command.																					
		<table border="1"> <thead> <tr> <th>Pre-Deblock Flag</th> <th>Post-Deblock Flag</th> <th>VD Pixels Output to Memory</th> <th>VD Pixels Output to SFC</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Invalid for SFC Mode</td> <td>Invalid for SFC Mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>Filtered Pixels (allow ON/OFF)</td> <td>Filter Pixels Sent to SFC for Scaling</td> </tr> <tr> <td>1</td> <td>0</td> <td>Non-filter (bypass) pixels (allow ON/OFF)</td> <td>Non-Filter Pixels Sent to SFC for Scaling</td> </tr> <tr> <td>1</td> <td>1</td> <td>Non-filter (bypass) pixels (allow ON/OFF)Filtered pixels (always OFF)</td> <td>Filter Pixels Sent to SFC for Scaling.</td> </tr> </tbody> </table>			Pre-Deblock Flag	Post-Deblock Flag	VD Pixels Output to Memory	VD Pixels Output to SFC	0	0	Invalid for SFC Mode	Invalid for SFC Mode	0	1	Filtered Pixels (allow ON/OFF)	Filter Pixels Sent to SFC for Scaling	1	0	Non-filter (bypass) pixels (allow ON/OFF)	Non-Filter Pixels Sent to SFC for Scaling	1	1	Non-filter (bypass) pixels (allow ON/OFF)Filtered pixels (always OFF)
Pre-Deblock Flag		Post-Deblock Flag	VD Pixels Output to Memory	VD Pixels Output to SFC																			
0		0	Invalid for SFC Mode	Invalid for SFC Mode																			
0		1	Filtered Pixels (allow ON/OFF)	Filter Pixels Sent to SFC for Scaling																			
1		0	Non-filter (bypass) pixels (allow ON/OFF)	Non-Filter Pixels Sent to SFC for Scaling																			
1		1	Non-filter (bypass) pixels (allow ON/OFF)Filtered pixels (always OFF)	Filter Pixels Sent to SFC for Scaling.																			
<b>VE - image enhanced pixel Output Enable</b>																							
For VE Mode, this field indicates if the VEBOX will enable writing out the image enhanced pixels to memory which is streamed to SFC pipeline for scaling. Filtered data is streamed directly from VEBOX to SFC through a dedicated internal interface. The pixel data send from VE to SFC is YUV format in 12-bit precision irrespective of VEBOX input surface type, pixel precision, chroma format, and color format (RGBA/YUVA). The following table shows allowed usage with VE -image enhanced pixel output enable along with SFC being enabled.																							
<table border="1"> <thead> <tr> <th>VE Output Surface Format</th> <th>Bits per channel</th> <th>Can SFC be enabled ?</th> </tr> </thead> <tbody> <tr> <td>Y8/ NV12/ AYUV/ YUYV/YVYU/UYYV/VYUY</td> <td>8bit</td> <td>Yes</td> </tr> <tr> <td>RGBA8</td> <td>8bit</td> <td>No</td> </tr> <tr> <td>RGBA10</td> <td>10bit</td> <td>No</td> </tr> <tr> <td>RGBA16</td> <td>16bit</td> <td>No</td> </tr> <tr> <td>Y16/ P216/P016/ Y216/ Y416</td> <td>16bit</td> <td>Yes</td> </tr> </tbody> </table>			VE Output Surface Format	Bits per channel	Can SFC be enabled ?	Y8/ NV12/ AYUV/ YUYV/YVYU/UYYV/VYUY	8bit	Yes	RGBA8	8bit	No	RGBA10	10bit	No	RGBA16	16bit	No	Y16/ P216/P016/ Y216/ Y416	16bit	Yes			
VE Output Surface Format	Bits per channel	Can SFC be enabled ?																					
Y8/ NV12/ AYUV/ YUYV/YVYU/UYYV/VYUY	8bit	Yes																					
RGBA8	8bit	No																					
RGBA10	10bit	No																					
RGBA16	16bit	No																					
Y16/ P216/P016/ Y216/ Y416	16bit	Yes																					



<b>SFC_LOCK_BODY</b>	
	<p>When DN is enabled, DN output is always on. When DI is Enabled, VE must send the first DI constructed surface to SFC in case VEBOX state indicate two DI frames output, while stream out the second DI constructed surface to memory. Else the DI output which is enabled will be sent out to SFC. VE output surface Type is programmed in VE_State command.</p>
0	<b>VE-SFC Pipe Select</b>

## SFC\_STATE\_BODY

<b>SFC_STATE_BODY</b>																		
Size (in bits):	1568																	
Default Value:	0x00000000, 0x00001000, 0x00000000																	
DWord	Bit	Description																
0	31:24	<b>Reserved</b>																
		Access:	RO															
		Format:	MBZ															
	23:18	<b>Reserved</b>																
		Access:	RO															
		Format:	MBZ															
	17:14	<b>Reserved</b>																
		Access:	RO															
		Format:	MBZ															
	13:12	<b>SFC Engine Mode</b>	Format:	U2														
			<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td></td> <td>Single SFC mode.</td> </tr> <tr> <td>01b</td> <td></td> <td>Left Most SFC in scalability/split-frame mode.</td> </tr> <tr> <td>10b</td> <td></td> <td>Right Most SFC in scalability/split-frame mode.</td> </tr> <tr> <td>11b</td> <td></td> <td>Middle SFC in scalability/split-frame mode.</td> </tr> </tbody> </table>		Value	Name	Description	00b		Single SFC mode.	01b		Left Most SFC in scalability/split-frame mode.	10b		Right Most SFC in scalability/split-frame mode.	11b	
		Value	Name	Description														
		00b		Single SFC mode.														
01b			Left Most SFC in scalability/split-frame mode.															
10b			Right Most SFC in scalability/split-frame mode.															
11b			Middle SFC in scalability/split-frame mode.															
<b>Programming Notes</b>																		
If SFC Pipe Mode is HCP-to-SFC and VE-to-SFC modes .Programmer need to ensure SFC Engine Mode bits programmed is the same as HCP_PIPE_MODE_SELECT command, Multi-Engine Modebits. For VE-SFC mode, this field has to be programmed according to the position of the split This Field is ignored for other SFC Pipe Mode.																		

<b>SFC_STATE_BODY</b>																																																																	
<b>Restriction</b>																																																																	
In case of scalability, the tile width has to be a minimum of 128.																																																																	
11	<b>Reserved</b>																																																																
	Access:	RO																																																															
	Format:	MBZ																																																															
10:8	<b>VD/VE Input Ordering Mode</b>																																																																
	Format:	U3																																																															
	<ul style="list-style-type: none"> <li>• VD mode: (SFC pipe mode set as "0")</li> <li>• VE mode: (pipe mode set as "1 and 4")</li> </ul>																																																																
	For values for each mode, please refer to the table below:																																																																
	<ul style="list-style-type: none"> <li>• HCP mode : SFC Pipe Mode set as "2"</li> </ul>																																																																
	For values for each mode, please refer to the table below:																																																																
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> <th style="text-align: center;">Programming Notes</th> <th style="text-align: center;">Exists If</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td></td> <td>16x16 block z-scan order - no shift</td> <td></td> <td>//VD Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>16x16 block z-scan order - 4 pixels shift upward</td> <td></td> <td>//VD Mode</td> </tr> <tr> <td style="text-align: center;">2</td> <td></td> <td>8x8 block jpeg z-scan order</td> <td></td> <td>//VD Mode</td> </tr> <tr> <td style="text-align: center;">3</td> <td></td> <td>16x16 block jpeg z-scan order</td> <td></td> <td>//VD Mode</td> </tr> <tr> <td style="text-align: center;">4</td> <td></td> <td>16x16 block VP8 row-scan order - no shift</td> <td></td> <td>//VD Mode</td> </tr> <tr> <td style="text-align: center;">5-7</td> <td></td> <td>Reserved</td> <td></td> <td>//VD Mode</td> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td>16x16 block HEVC Decoder row-scan order -4 pixel shift upward</td> <td></td> <td>//HCP Mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td>32x32 block HEVC Decoder row-scan order -4 pixel shift upward</td> <td></td> <td>//HCP Mode</td> </tr> <tr> <td style="text-align: center;">2</td> <td></td> <td>64x64 block HEVC Decoder row-scan order -4 pixel shift upward</td> <td></td> <td>//HCP Mode</td> </tr> <tr> <td style="text-align: center;">3</td> <td></td> <td>64x64 block VP9 Decoder row-scan order - 8 pixel shift upward</td> <td></td> <td>//HCP Mode</td> </tr> <tr> <td style="text-align: center;">[4-7]</td> <td></td> <td>Reserved</td> <td></td> <td>//HCP Mode</td> </tr> </tbody> </table>					Value	Name	Description	Programming Notes	Exists If	0		16x16 block z-scan order - no shift		//VD Mode	1		16x16 block z-scan order - 4 pixels shift upward		//VD Mode	2		8x8 block jpeg z-scan order		//VD Mode	3		16x16 block jpeg z-scan order		//VD Mode	4		16x16 block VP8 row-scan order - no shift		//VD Mode	5-7		Reserved		//VD Mode	0		16x16 block HEVC Decoder row-scan order -4 pixel shift upward		//HCP Mode	1		32x32 block HEVC Decoder row-scan order -4 pixel shift upward		//HCP Mode	2		64x64 block HEVC Decoder row-scan order -4 pixel shift upward		//HCP Mode	3		64x64 block VP9 Decoder row-scan order - 8 pixel shift upward		//HCP Mode	[4-7]		Reserved		//HCP Mode
	Value	Name	Description	Programming Notes	Exists If																																																												
	0		16x16 block z-scan order - no shift		//VD Mode																																																												
	1		16x16 block z-scan order - 4 pixels shift upward		//VD Mode																																																												
2		8x8 block jpeg z-scan order		//VD Mode																																																													
3		16x16 block jpeg z-scan order		//VD Mode																																																													
4		16x16 block VP8 row-scan order - no shift		//VD Mode																																																													
5-7		Reserved		//VD Mode																																																													
0		16x16 block HEVC Decoder row-scan order -4 pixel shift upward		//HCP Mode																																																													
1		32x32 block HEVC Decoder row-scan order -4 pixel shift upward		//HCP Mode																																																													
2		64x64 block HEVC Decoder row-scan order -4 pixel shift upward		//HCP Mode																																																													
3		64x64 block VP9 Decoder row-scan order - 8 pixel shift upward		//HCP Mode																																																													
[4-7]		Reserved		//HCP Mode																																																													

## SFC\_STATE\_BODY

0		8x4 block column order, 64 pixel column		//VE Mode
1		4x4 block column order, 64 pixel column		//VE Mode
[2-7]		Reserved		//VE Mode

### Programming Notes

This field shall be programmed according to video modes used in VDBOX. NOTE: SFC supports progressive input and output only (Interlaced/MBAFF is not supported).

Video Mode	Surface Format	SFC Input Chroma Sub-Sampling	VD/VE Input Ordering Mode
VC1 w/o LF and w/o OS Note: VC1 LF applies for either ILDB	420 (NV12)	1	0
VC1 w/ LF or w/ OS or w/ both Note: VC1 LF applies for either ILDB		INVALID with SFC	INVALID with SFC
AVC w/o LF	Monochrome	0	0
AVC w/o LF	420 (NV12)	1	0
AVC with LF	Monochrome	0	1
AVC/VP8 with LF	420 (NV12)	1	1
VP8 w/o LF	420 (NV12)	1	4
JPEG (YUV Interleaved)	Monochrome	0	2
JPEG (YUV Interleaved)	420	1	3
JPEG (YUV Interleaved)	422H_2Y	2	2
JPEG (YUV Interleaved)	422H_4Y	2	3
JPEG (YUV Interleaved)	444	4	2
JPEG (YUV Interleaved)	411	5	2

This field shall be programmed according to Image enhancement modes used in VEBOS.

VEBOX MODE	VEBOX Single Pipe Enable Bit	SFC Input Surface Format	SFC Input Chroma Sub Sampling	VD/VE Input Ordering Mode
1. DN/HP with RGB input	1	Monochrome	0	1
	1	420 (NV12)	1	1
2. Camera pipe (DM) enabled	1	422H	2	1
	1	444	4	1
3. IECF with FECSC, CCM, FGC filters enabled				
All other modes:	0	Monochrome	0	0

## SFC\_STATE\_BODY

(Legacy DN/DI/IECP features)	0	420 (NV12)	1	0
	0	422H	2	0
	0	444	4	0

This field shall be programmed according to video mode used in HCP. Note: SFC supports progressive input and output only (interlace/mbaff is not supported).

Video Mode	Surface Format	SFC Input Chroma Sub Sampling	VD/VE Input ordering mode
HEVC 16x16 LCU	420/422/444	1 / 2 / 4	0
HEVC 32x32 LCU	420/422/444	1 / 2 / 4	1
HEVC 64x64 LCU	420/422/444	1/ 2 / 4	2
VP9 64x64 LCU	420/444	1 / 4	3 / 4

### 7:4 SFC Input Chroma Sub-Sampling

Value	Name	Description
0	4:0:0	SFC to insert UV channels
1	4:2:0	
2	4:2:2 Horizontal	VD: 2:1:1
3	Reserved	
4	4:4:4 Progressive/Interleaved	

#### Programming Notes

This field shall be programmed according to video modes used in VDBOX. NOTE: SFC supports progressive input and output only (Interlaced/MBAFF is not supported).

Video Mode	Surface Format	SFC Input Chroma Sub-Sampling	VD/VE Input Ordering Mode
VC1 w/o LF and w/o OS Note: VC1 LF applies for either ILDB	420 (NV12)	1	0
VC1 w/ LF or w/ OS or w/ both Note: VC1 LF applies for either ILDB		INVALID with SFC	INVALID with SFC
AVC w/o LF	Monochrome	0	0
AVC w/o LF	420 (NV12)	1	0
AVC with LF	Monochrome	0	1
AVC/VP8 with LF	420 (NV12)	1	1
VP8 w/o LF	420 (NV12)	1	4
JPEG (YUV Interleaved)	Monochrome	0	2
JPEG (YUV Interleaved)	420	1	3

## SFC\_STATE\_BODY

		JPEG (YUV Interleaved)	422H_2Y	2	2
		JPEG (YUV Interleaved)	422H_4Y	2	3
		JPEG (YUV Interleaved)	444	4	2
		This field shall be programmed according to Image enhancement modes used in VEBOX.			
		VEBOX MODE	Surface Format	SFC Input Chroma Sub Sampling	VD/VE Input Ordering Mode
		Legacy DN/DI/IECP features	Monochrome	0	0
		Legacy DN/DI/IECP features	420 (NV12)	1	0
		Legacy DN/DI/IECP features	422H	2	0
		Legacy DN/DI/IECP features	444	4	0
		Capture/Camera pipe enabled(Demosaic)	Monochrome	0	1
		Capture/Camera pipe enabled(Demosaic)	420 (NV12)	1	1
		Capture/Camera pipe enabled(Demosaic)	422H	2	1
		Capture/Camera pipe enabled(Demosaic)	444	4	1
	3:0	<b>SFC Pipe Mode</b>			
		<b>Value</b>	<b>Name</b>	<b>Description</b>	
		0		VD-to-SFC AVS	
		1		VE-to-SFC AVS + IEF + Rotation	
		2		HCP-to-SFC AVS	
		3		Reserved	
		4		VE-to-SFC Integral Image	
		5			
		[6,15]		Reserved	
		<b>Programming Notes</b>			
		Note: for SFC Pipe mode set to VE-to-SFC AVS mode. IECP pipeline mode MUST be enabled. However, each sub-IECP feature can be turned on/off independently.			
1	31:30	<b>Reserved</b>			
		Access:		RO	
		Format:		MBZ	
	29:16	<b>Input Frame Resolution Height</b>			
		Format:		U14-1	
		Minus 1 in unit of pixel [13:0]. It is set to the value of the output resolution or number of pixels			

## SFC\_STATE\_BODY

			<p>streaming into SFC from VD/HCP or VEBOX. Since the Max value support in 16K pixels, the max value allowed in 16K minus 1.</p> <ul style="list-style-type: none"> <li>• VDBOX frame height is multiple of 16 for Video source and JPEG formats other than 400, 444 and 422H_2Y.</li> <li>• VDBOX frame height is multiple of 8 for JPEG formats 400, 444 and 422H_2Y.</li> <li>• VEBOX frame height is multiple of 4.</li> <li>• HEVC frame height is multiple of 8</li> <li>• VP9 frame height is multiple of 8.</li> </ul> <p>Min Resolution is 32 pixels. Max Resolution is up to 16K pixel eg. for 1920x1080 content, FrameHeightInMBsMinus1 is equal to 1087 (1080 rounded up 16 pixel boundary, minus 1. i.e. effectively specified as 1088 instead).</p> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p>
	15:14	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	13:0	<b>Input Frame Resolution Width</b>	
		Format:	U14-1
		<p>Minus 1 in unit of pixel [13:0]. It is set to the value of the output resolution or number of pixels streaming into SFC from VD/HCP or VEBox. Since the max value support is 16k pixels, the max value allowed is 16K minus 1.</p> <ul style="list-style-type: none"> <li>• VDBOX frame width is multiple of 16 for Video source and JPEG formats other than 400, 444 and 422H_2Y.</li> <li>• VDBOX frame width is multiple of 8 for JPEG formats 400, 444 and 422H_2Y.</li> <li>• VEBOX frame width is multiple of 16.</li> <li>• HEVC frame width is multiple of 8.</li> <li>• VP9 frame width is multiple of 8.</li> </ul> <p><i>Min Resolution</i> is 32 pixels. <i>Max Resolution</i> is up to 16K pixels. e.g. for 1920x1080 content, FrameHeightInMBsMinus1 is equal to 1087 (1080 rounded up 16 pixel boundary, minus 1. i.e. effectively specified as 1088 instead).</p>	
		<b>Restriction</b>	
		For Integral Image Mode, this field is Reserved and MBZ.	
2	31:24	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ

## SFC\_STATE\_BODY

	23:17	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	16	<b>Input Color Space - 0- YUV/1 - RGB</b>	This specifies the color space of the input format. RGB is valid only with the VE-SFC mode.
		<b>Value</b>	<b>Name</b>
		0	YUV Color Space
		1	RGB Color Space
	15:12	<b>Output Chroma Downsampling co-siting position Horizontal Direction</b>	
		Format:	U4
		This field specifies the fractional position of the bilinear filter for chroma downsampling. In the X-axis.	
		<b>Value</b>	<b>Name</b>
		0000b	0 (fraction_in_integer)
		0001b	1 (fraction_in_integer)
		0010b	2 (fraction_in_integer)
		0011b	3 (fraction_in_integer)
		0100b	4 (fraction_in_integer)
		0101b	5 (fraction_in_integer)
		0110b	6 (fraction_in_integer)
		0111b	7 (fraction_in_integer)
		1000b	8/8
		<b>Programming Notes</b>	
		For 444 format, horizontal chroma-siting should be programmed to zero.	
	11:8	<b>Output Chroma Downsampling co-siting position Vertical Direction</b>	
		Format:	U4
		This field specifies the fractional position of the bilinear filter for chroma downsampling. In the Y-axis.	
		<b>Value</b>	<b>Name</b>
		0000b	0 (fraction_in_integer)
		0001b	1 (fraction_in_integer)
		0010b	2 (fraction_in_integer)
		0011b	3 (fraction_in_integer)
		0100b	4 (fraction_in_integer)
		0101b	5 (fraction_in_integer)
		0110b	6 (fraction_in_integer)
		0111b	7 (fraction_in_integer)



## SFC\_STATE\_BODY

		1000b	8/8	
		<b>Programming Notes</b>		
		For 444 and 422 format, vertical chroma co-siting value should be programmed to zero.		
7:6	<b>Reserved</b>			
	Access:	RO		
	Format:	MBZ		
5	<b>RGBA_Channel_Swap Enable</b>			
	Default Value:	0		
	Format:	Enable		
	This bit should only be used with RGB output formats and CSC conversion is turned on. When this bit is set, the R and B channels are swapped into the output RGB channels as shown in the following table:			
	<b>Name</b>	<b>Bits</b>	<b>MSB Color Order</b>	<b>Swapped</b>
	RGBA8	8:8:8:8	A:B:G:R	A:R:G:B
	RGBA10	2:10:10:10	A:R:G:B	A:B:G:R
	RGB 5:6:5	5:6:5	R:G:B	B:G:R
4	<b>Reserved</b>			
	Access:	RO		
	Format:	MBZ		
3:0	<b>Output Surface Format type</b>			
	SFC output surface format type.			
	Reserved			
	<b>Value</b>	<b>Name</b>	<b>Description</b>	<b>Exists If</b>
	0		AYUV 4:4:4 (8:8:8:8 MSB-A:Y:U:V)	//Tile-Y/ Tile-X/Linear
	1		RGBA8 4:4:4:4 (8:8:8:8 MSB-A:B:G:R)	//Tile-Y/ Tile-X/Linear
	2		RGBA10 10:10:10:2 (2:10:10:10 MSB-A:R:G:B)	//Tile-Y/ Tile-X/Linear
	3		RGB 5:6:5 (5:6:5 MSB-R:G:B)	//Tile-Y/ Tile-X/Linear
	4		Planar NV12 4:2:0 8-bit	//Tile-Y
	5		Packed YUYV 4:2:2 8-bit	//Tile-Y/ Tile-X/Linear
	6		Packed UYVY 4:2:2 8-bit	//Tile-Y/ Tile-X/Linear
	7		Packed integral Image 32-bit	//Linear
	8		Packed integral Image 64-bit	//Linear
	9		P016 format	//Tile-Y
	10		Y210 / Y216 Format BitDepth = 0 => Y210 BitDepth = 1 => Y216	//Tile-Y / Tile-X / Linear

<b>SFC_STATE_BODY</b>			
	11	Y410 / Y416 Format BitDepth = 0 => Y410 BitDepth = 1 => Y416	//Tile-Y / Tile-X / Linear
		<b>Restriction</b>	
		For Integral Image Mode, output surface format type must be set to 32/64-bit Integral Image Plane. Driver/SW must ensure the max accumulated integral image value does not exceed the programmable output precision. HW will simply generate wrong value once it overflow in wrap around case.	
3	31	<b>Reserved</b>	
	30	Access:	RO
		Format:	MBZ
	29:24	Access:	RO
		Format:	MBZ
	23	<b>Histogram Streamout</b>	
		Format:	Boolean
	22	<b>Tile Type</b>	
		Format:	Boolean
		0 : Real HCP Tile Mode 1 : Virtual HCP Tile Mode	
<b>Programming Notes</b>			
This field is only used when SFC Pipe Mode is HCP-to-SFC. In Real HCP Tile Mode, video streams defines the tile boundary. In Virtual HCP Tile Mode, driverstreams defines the tile boundary.			
21:20	<b>BitDepth</b> This field is valid only for output formats P016/Y216/Y416. This field is used to specify how many of the LSB bits have valid data.		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0	10BitFormat	Higher 10 bits are valid and lower 6 bits are 0
19	<b>CSC Enable</b> This field is set when YUV to RGB or RGB to YUV conversion is required or the RGB/YUV range conversion is required. CSC conversion matrix need to be programmed accordingly.		
	<b>Restriction</b>		
	For Integral Image Mode, this field is Reserved and MBZ.		

## SFC\_STATE\_BODY

18	<b>Color Fill Enable</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center; background-color: #e1eef6;">Programming Notes</th> </tr> <tr> <td colspan="2">This field could be enabled only if the scaled resolution is smaller than the output/display resolution. If enabled, HW will fill the gap with programmable pixel values. Else, nothing will be filled in the gap region.</td> </tr> <tr> <td colspan="2"><b>Usage:</b> Color fill must be enabled for the first time/pass when a new surface is allocated/ used. Optional for subsequence frames since the gap region is filled with default pixels by prior passes.</td> </tr> <tr> <td colspan="2">In scalability mode ie. (SFC Engine Mode != 00), gray fill should be set only for left mosttile and for other tiles it should be disabled.</td> </tr> </table>		Programming Notes		This field could be enabled only if the scaled resolution is smaller than the output/display resolution. If enabled, HW will fill the gap with programmable pixel values. Else, nothing will be filled in the gap region.		<b>Usage:</b> Color fill must be enabled for the first time/pass when a new surface is allocated/ used. Optional for subsequence frames since the gap region is filled with default pixels by prior passes.		In scalability mode ie. (SFC Engine Mode != 00), gray fill should be set only for left mosttile and for other tiles it should be disabled.													
Programming Notes																							
This field could be enabled only if the scaled resolution is smaller than the output/display resolution. If enabled, HW will fill the gap with programmable pixel values. Else, nothing will be filled in the gap region.																							
<b>Usage:</b> Color fill must be enabled for the first time/pass when a new surface is allocated/ used. Optional for subsequence frames since the gap region is filled with default pixels by prior passes.																							
In scalability mode ie. (SFC Engine Mode != 00), gray fill should be set only for left mosttile and for other tiles it should be disabled.																							
17:16	<b>Rotation Mode</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U2</td> </tr> <tr> <th style="width: 30%; text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>00b</td> <td>0 (degrees)</td> </tr> <tr> <td>01b</td> <td>90 Clockwise</td> </tr> <tr> <td>10b</td> <td>180 Clockwise</td> </tr> <tr> <td>11b</td> <td>270 Clockwise</td> </tr> <tr> <th colspan="2" style="text-align: center; background-color: #e1eef6;">Programming Notes</th> </tr> <tr> <td colspan="2">SFC rotation (90, 180 and 270) should be set only on VEBox input mode and SFC output set to TileY.</td> </tr> <tr> <td colspan="2">Restriction:</td> </tr> <tr> <td colspan="2"> <ul style="list-style-type: none"> <li>For Integral Image Mode, this field is Reserved and MBZ.</li> <li>For VDBox Mode, this field is Reserved and MBZ.</li> <li>For linear or TileX SFC output, this field is Reserved and MBZ.</li> </ul> </td> </tr> </table>		Format:	U2	Value	Name	00b	0 (degrees)	01b	90 Clockwise	10b	180 Clockwise	11b	270 Clockwise	Programming Notes		SFC rotation (90, 180 and 270) should be set only on VEBox input mode and SFC output set to TileY.		Restriction:		<ul style="list-style-type: none"> <li>For Integral Image Mode, this field is Reserved and MBZ.</li> <li>For VDBox Mode, this field is Reserved and MBZ.</li> <li>For linear or TileX SFC output, this field is Reserved and MBZ.</li> </ul>	
Format:	U2																						
Value	Name																						
00b	0 (degrees)																						
01b	90 Clockwise																						
10b	180 Clockwise																						
11b	270 Clockwise																						
Programming Notes																							
SFC rotation (90, 180 and 270) should be set only on VEBox input mode and SFC output set to TileY.																							
Restriction:																							
<ul style="list-style-type: none"> <li>For Integral Image Mode, this field is Reserved and MBZ.</li> <li>For VDBox Mode, this field is Reserved and MBZ.</li> <li>For linear or TileX SFC output, this field is Reserved and MBZ.</li> </ul>																							
15	<b>Reserved</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ																
Access:	RO																						
Format:	MBZ																						
14	<b>Mirror mode</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Format:</td> <td style="width: 50%;">Boolean</td> </tr> <tr> <th style="width: 25%; text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>0</td> <td>Mirror mode Disabled</td> </tr> <tr> <td>1</td> <td>Mirror mode Enabled</td> </tr> <tr> <th colspan="2" style="text-align: center; background-color: #e1eef6;">Programming Notes</th> </tr> <tr> <td colspan="2">0 : Mirror Mode disabled</td> </tr> </table>		Format:	Boolean	Value	Name	0	Mirror mode Disabled	1	Mirror mode Enabled	Programming Notes		0 : Mirror Mode disabled									
Format:	Boolean																						
Value	Name																						
0	Mirror mode Disabled																						
1	Mirror mode Enabled																						
Programming Notes																							
0 : Mirror Mode disabled																							

<b>SFC_STATE_BODY</b>		
	1 : Mirror Mode enabled	
13	<b>Mirror type</b>	
	Format:	Boolean
	<b>Value</b>	<b>Name</b>
	0	Horizontal flip
	1	Vertical flip
<b>Programming Notes</b>		
0 : Horizontal flip		
1 : Vertical flip		
<b>Restriction</b>		
When Mirror mode is set to 0, this field should be programmed to 0.		
12	<b>Chroma Upsampling Enable</b>	
	This field enables the high-quality UV channel upsampler prior to IEF filter process. This field should be disabled when the source pixels and output pixels are kept with the same chroma sub-sample type and IEF is disabled.	
<b>Restriction</b>		
For Integral Image Mode, this field is Reserved and MBZ.		
11	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
10	<b>RGB Adaptive</b>	
	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.	
	0: Disable the RGB Adaptive equation and use G-Ch directly for adaptive filter	
	1: Enable the RGB Adaptive filter using the equation $(Y=(R+2G+B)\gg 2)$	
9	<b>Bypass X Adaptive Filtering</b>	
	<b>Value</b>	<b>Name</b>
	<b>Description</b>	
	0	Enable X Adaptive Filtering
	1	Disable X Adaptive Filtering
	The X direction will use <b>Default Sharpness Level</b> to blend between the smooth and sharp filters rather than the calculated value.	
8	<b>Bypass Y Adaptive Filtering</b>	
	<b>Value</b>	<b>Name</b>
	<b>Description</b>	
	0	Enable Y Adaptive Filtering

## SFC\_STATE\_BODY

	1	Disable Y Adaptive Filtering	The Y direction will use <b>Default Sharpness Level</b> to blend between the smooth and sharp filters rather than the calculated value.
7	<b>AVS Scaling Enable</b>		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	1	Enable	
	0	Disable	The scaling factor is ignored and a scaling ratio of 1:1 is assumed.
6	<b>Adaptive Filter for all Channels</b>		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	1	Enable Adaptive Filter on UV/RB Channels	8-tap Adaptive Filter Mode is on
	0	Disable Adaptive Filter on UV/RB Channels	
	<b>Programming Notes</b>		
	The field can be enabled if 8-tap Adaptive filter mode is on. Else it should be disabled.		
5:4	<b>AVS Filter Mode</b>		
	<b>Value</b>	<b>Name</b>	
	0	5x5 Poly-phase filter + Bilinear (adaptive)	
	1	8x8 poly-phase filter + Bilinear (adaptive)	
	2	Bilinear filter only	
	3	Reserved	
	<b>Programming Notes</b>		
	In VD-to-SFC mode, value of 1 is not allowed.		
3	<b>Enable 8 tap for Chroma channels filtering</b>		
	This bit enables 8 tap filtering for Chroma Channels.		
	<b>Programming Notes</b>		
	8tap enable should only be enabled when SFC Input Chroma Sub-Sampling = 4 (ie. 444 input format to SFC).		
2	<b>IEF4Smooth_Enable</b>		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	0	<b>[Default]</b>	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
	<b>Restriction</b>		
	For Integral Image Mode, this field is Reserved and MBZ.		
1	<b>Skin Tone Tuned IEF_Enable</b>		
	Exists If:		//IEF Enable = 1

<b>SFC_STATE_BODY</b>											
		<p style="text-align: center;"><b>Restriction</b></p> <p>For Integral Image Mode, this field is Reserved and MBZ.</p>									
	0	<p><b>IEF Enable</b></p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>Enable</td> <td>IEF Filter is Enabled</td> </tr> <tr> <td style="text-align: center;">0</td> <td>Disable</td> <td>IEF Filter is Disabled</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Restriction</b></p> <p>For Integral Image Mode and VD Mode, this field is Reserved and MBZ.</p>	Value	Name	Description	1	Enable	IEF Filter is Enabled	0	Disable	IEF Filter is Disabled
Value	Name	Description									
1	Enable	IEF Filter is Enabled									
0	Disable	IEF Filter is Disabled									
4	31:30	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO										
Format:	MBZ										
	29:16	<p><b>Source Region Height</b></p> <table border="1"> <tr> <td>Format:</td> <td>U14-1</td> </tr> </table> <p><b>Source/Crop Region Height Minus 1 of the Input Frame in Unit of Pixel [13:0].</b>  This field specifies the source/crop region of the input frame used for scaling of the graphic view. It defines the out-of-frame boundary used prior to AVS/IEF interpolation operation. The max value should be programmed to be equal or small than the input FrameHeightInMBminus 1 field.  e.g. for 1920x1080 content, FrameHeightInMBsMinus1 is equal to 1087 (1088 lines); however, the crop region height should be set to 1079(1080 lines). The last 8 lines are assumed to be not usable and should not be used as source pixels for Scaling or IEF operations. Otherwise, the bad pixels will breach and cause artifacts into the scaled output frame.</p> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p> <p>Restriction : For AVS mode, the restriction is tied to chroma input format type: 420 - multiple of 2. 422/444/400 - no restrictions, except for AVS bypass case (ie. 1:1 scaling) where restriction is tied to chroma output format. <b>Min Resolution</b> is 32 pixels. <b>Max Resolution</b> is 16K pixels.</p> <p style="text-align: center;"><b>Restriction</b></p> <p>In VD-to-SFC and HCP-to-SFC modes, this field must be programmed to same value as Input Frame Resolution Height.</p>	Format:	U14-1							
Format:	U14-1										
	15:14	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO										
Format:	MBZ										
	13:0	<p><b>Source Region Width</b></p> <table border="1"> <tr> <td>Format:</td> <td>U14-1</td> </tr> </table> <p><b>Source/Crop Region Width Minus 1 of the Input Frame in Unit of Pixel [13:0].</b>  This field specifies the source/crop region of the input frame used for scaling of the graphic</p>	Format:	U14-1							
Format:	U14-1										

## SFC\_STATE\_BODY

		<p>view. It defines the out-of-frame boundary used prior to AVS/IEF interpolation operation. The max value should be programmed to be equal or small than the input FrameWidthInMBminus 1 field.</p> <p>e.g. for 1920x1080 content, FrameWidthInMBsMinus1 is equal to 1919 (1920 pixel wide); however, the crop region width should be set to less than 1909(1910 pixel wide). The last 10 pixels of the frame are assumed to be not usable and should not be used as source pixels for Scaling or IEF operations. Otherwise, the bad pixels will breach and cause artifacts into the scaled output frame.</p> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p> <p>Restriction : For AVS mode, the restriction is tied to chroma input format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions, except for AVS bypass case (ie. 1:1 scaling) where restriction is tied to chroma output format. <b>Min Resolution</b> is 32 pixels. <b>Max Resolution</b> is 16K pixels.</p>					
		<b>Restriction</b>					
		<p>In VD-to-SFC and HCP-to-SFC modes, this field must be programmed to same value as Input Frame Resolution Width.</p>					
5	31:30	<p><b>Reserved</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ
Access:	RO						
Format:	MBZ						
	29:16	<p><b>Source Region Vertical Offset</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td>U14</td> </tr> </table> <p><b>Vertical Offset Of The SRC Region Relative To The Starting Position Of The Input Frame In Unit Of Pixel [13:0]</b></p> <p>This field specifies the vertical offset of the starting position of the scaled region relatives to the starting position (pixel 0,0) of the output frame. It defines the out-of-frame boundary used prior to AVS/IEF interpolation operation. This value should be set to zero if the starting corner of the crop region is same as the input frame region. The sum of this value and the src/crop region size heightminus1 must be programmed to be equal or small than the input FrameHeightinMBminus 1 field.</p> <p>Restriction : For Integral Image Mode, this field is Reserved and MBZ.</p> <p>Restriction : For AVS mode, the restriction is tied to chroma input format type: 420 - multiple of 2. 422/444/400 - no restrictions.</p>		Format:	U14		
Format:	U14						
		<b>Restriction</b>					
		<p>In VD-to-SFC and HCP-to-SFC modes, this field is Reserved and MBZ..</p>					
	15:14	<p><b>Reserved</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ
Access:	RO						
Format:	MBZ						

<b>SFC_STATE_BODY</b>		
	13:0	<b>Source Region Horizontal Offset</b>
		Format: U14
		<b>Horizontal Offset Of The SRC Region Relative To The Starting Position Of The Input Frame In Unit Of Pixel [13:0]</b> This field specifies the horizontal offset of the starting position of the scaled region relatives to the starting position (pixel 0,0) of the output frame. It defines the out-of-frame boundary used prior to AVS/IEF interpolation operation. This value should be set to zero if the starting corner of the crop region is same as the input frame region. The sum of this value and the src/crop region size widthminus1 must be programmed to be equal or small than the input FrameWidthinMBminus 1 field.
		Restriction : For Integral Image Mode, this field is Reserved and MBZ. Restriction : For AVS mode, the restriction is tied to chroma input format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions.
		<b>Restriction</b> In VD-to-SFC and HCP-to-SFC modes, this field is Reserved and MBZ..
6	31:30	<b>Reserved</b>
		Access: RO
		Format: MBZ
	29:16	<b>Output Frame Height</b>
		Format: U14-1
		It is set to the value of the final output resolution of the graphic view. Since the max value support is 16k pixels, the max value allowed is 16K minus 1.
		Restriction : For Integral Image Mode, this field is Reserved and MBZ. Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions. <b>Min Resolution</b> is 32 pixels. <b>Max Resolution</b> is 16K pixels.
	15:14	<b>Reserved</b>
		Access: RO
		Format: MBZ
	13:0	<b>Output Frame Width</b>
		Format: U14-1
		It is set to the value of the final output resolution of the graphic view. Since the max value support is 16k pixels, the max value allowed is 16K minus 1.
		Restriction : For Integral Image Mode, this field is Reserved and MBZ. Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions. <b>Min Resolution</b> is 32 pixels. <b>Max Resolution</b> is 16K pixels.



<b>SFC_STATE_BODY</b>						
7	31:30	<b>Reserved</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	29:16	<b>Scaled Region Size Height</b>				
		Format: U14-1				
		It is set to the height of the scaled region over the output frame of the graphic view.				
		Restriction : For AVS mode, if rotation_mode = 0/180, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions. For AVS mode, if rotation_mode = 90/270, the restriction is tied to chroma output format type: 420/422 - multiple of 2. 444/400 - no restrictions. <b>Min Resolution</b> is 32 pixels. <b>Max Resolution</b> is 16K pixels.				
		<b>Programming Notes</b>				
		The Max Value = < [The Output Frame Height Minus1].				
<b>Restriction</b>						
In VD-to-SFC, HCP-to-SFC and AV1-to-SFCmodes, this field should be programmed to a value of Output Frame Height Minus1						
15:14	<b>Reserved</b>					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
13:0	<b>Scaled Region Size Width</b>					
	Format: U14-1					
	It is set to the Width of the scaled region over the output frame of the graphic view.					
	Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions. <b>Min Resolution</b> is 32 pixels. <b>Max Resolution</b> is 16K pixels					
	<b>Programming Notes</b>					
	The Max Value = < [The Output Frame Width Minus1].					
	<b>Restriction</b>					
In VD-to-SFC, HCP-to-SFC and AV1-to-SFCmodes, this field should be programmed to a value of Output Frame Width Minus1						
8	31	<b>Reserved</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
		Access:	RO			
Format:	MBZ					

## SFC\_STATE\_BODY

	30:16	<b>Scaled Region Vertical Offset</b>	Format:	S14	
			<p><b>Vertical Offset (in pixels) Of The Scaled Region Relatives to The Starting Position Of The Output Frame In Unit Of Pixel [13:0]</b></p> <p>This field specifies the vertical offset of the starting position of the scaled region relatives to the starting position (pixel 0,0) of the output frame. The gap between the scaled and output frame shall be filled by hardware with a set of programmed YUV/RGB values (Grey Bar). This value should be set to zero if the starting corner of the scaled region is same as the output frame region. The sum of this value and the scaled region size Heightminus1 must be programmed to be equal or small than the output FrameHeightinMBminus 1 field plus 16.</p>		
			<b>Restriction</b>		
			For Integral Mode, this field is reserved and MBZ		
			For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422/444/400 - no restrictions.		
			In VD-to-SFC, HCP-to-SFC and AV1-to-SFC modes, this field is Reserved and MBZ..		
	15	<b>Reserved</b>	Access:	RO	
			Format:	MBZ	
	14:0	<b>Scaled Region Horizontal Offset</b>	Format:	S14	
			<p><b>Horizontal Offset (in pixels) Of The Scaled Region Relatives to The Starting Position Of The Output Frame In Unit Of Pixel [13:0]</b></p> <p>This field specifies the horizontal offset of the starting position of the scaled region relatives to the starting position (pixel 0,0) of the output frame. The gap between the scaled and output frame shall be filled by hardware with a set of programmed YUV/RGB values (Grey Bar). This value should be set to zero if the starting corner of the scaled region is same as the output frame region. The sum of this value and the scaled region size Widthminus1 must be programmed to be equal or small than the output FrameWidthinMBminus 1 field plus 16.</p>		
			Restriction : For Integral Image Mode, this field is Reserved and MBZ.		
			Restriction : For AVS mode, the restriction is tied to chroma output format type: 420 - multiple of 2. 422 - multiple of 2. 444/400 - no restrictions.		
			<b>Restriction</b>		
			In VD-to-SFC, HCP-to-SFC and AV1-to-SFC modes, this field is Reserved and MBZ..		
9	31:26	<b>Reserved</b>	Access:	RO	
			Format:	MBZ	
	25:16	<b>Gray Bar Pixel - Y/R</b>	Format:	U1.9	

<b>SFC_STATE_BODY</b>																																	
	<table border="1"> <tr> <td colspan="2"><b>Range:</b>[0.0, +1.0]</td> </tr> <tr> <td colspan="2">This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in Y or R channel on the AYUV or RGBA domain respectively.</td> </tr> <tr> <td colspan="2" style="text-align: center;"><b>Restriction</b></td> </tr> <tr> <td colspan="2">For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	<b>Range:</b> [0.0, +1.0]		This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in Y or R channel on the AYUV or RGBA domain respectively.		<b>Restriction</b>		For Integral Image Mode, this field is Reserved and MBZ.																									
<b>Range:</b> [0.0, +1.0]																																	
This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in Y or R channel on the AYUV or RGBA domain respectively.																																	
<b>Restriction</b>																																	
For Integral Image Mode, this field is Reserved and MBZ.																																	
	<table border="1"> <tr> <td><b>15:10</b></td> <td><b>Reserved</b></td> </tr> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	<b>15:10</b>	<b>Reserved</b>	Access:	RO	Format:	MBZ																										
<b>15:10</b>	<b>Reserved</b>																																
Access:	RO																																
Format:	MBZ																																
	<table border="1"> <tr> <td><b>9:0</b></td> <td><b>Gray Bar Pixel - U/G</b></td> </tr> <tr> <td>Format:</td> <td>U1.9</td> </tr> <tr> <td colspan="2"><b>Range:</b>[0.0, +1.0]</td> </tr> <tr> <td colspan="2">This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in U or G channel on the AYUV or RGBA domain respectively.</td> </tr> <tr> <td colspan="2" style="text-align: center;"><b>Restriction</b></td> </tr> <tr> <td colspan="2">For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	<b>9:0</b>	<b>Gray Bar Pixel - U/G</b>	Format:	U1.9	<b>Range:</b> [0.0, +1.0]		This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in U or G channel on the AYUV or RGBA domain respectively.		<b>Restriction</b>		For Integral Image Mode, this field is Reserved and MBZ.																					
<b>9:0</b>	<b>Gray Bar Pixel - U/G</b>																																
Format:	U1.9																																
<b>Range:</b> [0.0, +1.0]																																	
This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in U or G channel on the AYUV or RGBA domain respectively.																																	
<b>Restriction</b>																																	
For Integral Image Mode, this field is Reserved and MBZ.																																	
10	<table border="1"> <tr> <td><b>31:26</b></td> <td><b>Reserved</b></td> </tr> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> <tr> <td><b>25:16</b></td> <td><b>Gray Bar Pixel - V/B</b></td> </tr> <tr> <td>Format:</td> <td>U1.9</td> </tr> <tr> <td colspan="2"><b>Range:</b>[0.0, +1.0]</td> </tr> <tr> <td colspan="2">This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in V or B channel on the AYUV or RGBA domain respectively.</td> </tr> <tr> <td colspan="2" style="text-align: center;"><b>Restriction</b></td> </tr> <tr> <td colspan="2">For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> <tr> <td><b>15:10</b></td> <td><b>Reserved</b></td> </tr> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> <tr> <td><b>9:0</b></td> <td><b>Gray Bar Pixel - A</b></td> </tr> <tr> <td>Format:</td> <td>U1.9</td> </tr> <tr> <td colspan="2"><b>Range:</b>[0.0, +1.0]</td> </tr> <tr> <td colspan="2">This is the default value used to fill in the area between the scaled region and the output frame</td> </tr> </table>	<b>31:26</b>	<b>Reserved</b>	Access:	RO	Format:	MBZ	<b>25:16</b>	<b>Gray Bar Pixel - V/B</b>	Format:	U1.9	<b>Range:</b> [0.0, +1.0]		This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in V or B channel on the AYUV or RGBA domain respectively.		<b>Restriction</b>		For Integral Image Mode, this field is Reserved and MBZ.		<b>15:10</b>	<b>Reserved</b>	Access:	RO	Format:	MBZ	<b>9:0</b>	<b>Gray Bar Pixel - A</b>	Format:	U1.9	<b>Range:</b> [0.0, +1.0]		This is the default value used to fill in the area between the scaled region and the output frame	
<b>31:26</b>	<b>Reserved</b>																																
Access:	RO																																
Format:	MBZ																																
<b>25:16</b>	<b>Gray Bar Pixel - V/B</b>																																
Format:	U1.9																																
<b>Range:</b> [0.0, +1.0]																																	
This is the default value used to fill in the area between the scaled region and the output frame size (aka Gray Bar) in V or B channel on the AYUV or RGBA domain respectively.																																	
<b>Restriction</b>																																	
For Integral Image Mode, this field is Reserved and MBZ.																																	
<b>15:10</b>	<b>Reserved</b>																																
Access:	RO																																
Format:	MBZ																																
<b>9:0</b>	<b>Gray Bar Pixel - A</b>																																
Format:	U1.9																																
<b>Range:</b> [0.0, +1.0]																																	
This is the default value used to fill in the area between the scaled region and the output frame																																	

<b>SFC_STATE_BODY</b>					
	<p>size (aka Gray Bar) in A channel on the AYUV or RGBA domain respectively.</p> <p style="text-align: center;"><b>Restriction</b></p> <p>For Integral Image Mode, this field is Reserved and MBZ.</p>				
11	<p><b>31:26 Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO			
	Format:	MBZ			
	<p><b>25:16 UV Default value for V channel (For Mono Input Support)</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Exists If:</td> <td>//Input NOT originated by VEBOX.</td> </tr> <tr> <td>Format:</td> <td>U1.9</td> </tr> </table> <p><b>Range:</b>[0.0, +1.0]</p> <p>This field specifies the UV default value fill in to the UV output channels when input from VDBOX is set to Monochrome.</p> <p style="text-align: center;"><b>Restriction</b></p> <p>Not used when input is originated by VEBOX (Including Integral Image Mode).</p>	Exists If:	//Input NOT originated by VEBOX.	Format:	U1.9
	Exists If:	//Input NOT originated by VEBOX.			
	Format:	U1.9			
	<p><b>15:10 Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO			
	Format:	MBZ			
	<p><b>9:0 UV Default value for U channel (For Mono Input Support)</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Exists If:</td> <td>//Input NOT originated by VEBOX.</td> </tr> <tr> <td>Format:</td> <td>U1.9</td> </tr> </table> <p><b>Range:</b>[0.0, +1.0]</p> <p>This field specifies the UV default value fill in to the UV output channels when input from VDBOX is set to Monochrome.</p> <p style="text-align: center;"><b>Restriction</b></p> <p>Not used when input is originated by VEBOX (Including Integral Image Mode).</p>	Exists If:	//Input NOT originated by VEBOX.	Format:	U1.9
Exists If:	//Input NOT originated by VEBOX.				
Format:	U1.9				
12	<p><b>31:10 Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO			
	Format:	MBZ			
<p><b>9:0 Alpha Default Value</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U1.9</td> </tr> </table> <p><b>Range:</b>[0.0, +1.0]</p> <p>This field specifies the Alpha default value fill into the alpha output channel when output</p>	Format:	U1.9			
Format:	U1.9				

<b>SFC_STATE_BODY</b>						
		<p>format type is set to RGBA8/10.</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="2"><b>Restriction</b></td> </tr> <tr> <td colspan="2">For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	<b>Restriction</b>		For Integral Image Mode, this field is Reserved and MBZ.	
<b>Restriction</b>						
For Integral Image Mode, this field is Reserved and MBZ.						
13	31:28	<b>Reserved</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
27:5	<b>Scaling Factor Height</b>					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U4.19</td> </tr> </table> <p>This field specifies the scaling ratio of the vertical sizes between the crop/source region and the scaled region. The destination pixel coordinate, y-axis, is multiplied with this scaling factor to mapping back to the source input pixel coordinate.</p> <p>The field specifies the ratio of crop height resolution/ scaled height resolution. This implies <math>1/sf_v</math> in the equation.</p>	Format:	U4.19			
Format:	U4.19					
4:0	<b>Reserved</b>					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
14	31:28	<b>Reserved</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
27:5	<b>Scale Factor Width</b>					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U4.19</td> </tr> </table> <p>This field specifies the scaling ratio of the horizontal sizes between the crop/source region and the scaled region. The destination pixel coordinate, x-axis, is multiplied with this scaling factor to mapping back to the source input pixel coordinate.</p> <p>The field specifies the ratio of crop width resolution/ scaled width resolution. This implies <math>1/sf_u</math> in the equations above.</p>	Format:	U4.19			
Format:	U4.19					
4:0	<b>Reserved</b>					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
15	31:22	<b>Reserved</b>				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
21:0	<b>Reserved</b>					
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					

<b>SFC_STATE_BODY</b>				
16	31:12	<b>Output Frame Surface Base Address</b>		
		Specifies the 4K byte aligned frame buffer address for outputting the scaled up/down image. Data is stored in Tile-Y format.		
		For Integral Image mode, the accumulated integral image values will be packed linear in this surface.		
		<b>Programming Notes</b>		
		This field is ignored if I-frame only mode is set to 0 (Disable).		
11:0	<b>Reserved</b>	Access: RO		
		Format: MBZ		
17	31:16	<b>Reserved</b>		
		Access: RO		
	Format: MBZ			
	15:0	<b>Output Frame Surface Base Address High</b>		
This field is for the upper range [47:32] of Output Frame Surface Base Address. For Integral Image mode, the accumulated integral image values will be packed linear in this surface.				
18	31:15	<b>Reserved</b>		
		Access: RO		
	Format: MBZ			
	14:13	<b>Output Surface Tiled Mode</b>		
		Format: U2		
		<b>For Media Surfaces:</b> This field specifies the tiled resource mode.		
		<b>Value</b>	<b>Name</b>	<b>Description</b>
		0h	TRMODE_NONE	No tiled resource
		1h	TRMODE_TILEYF	Reserved
		2h	TRMODE_TILEYS	reserved
3h	Reserved			
12	<b>Output Frame Surface Base Address - Row Store Scratch Buffer Cache Select</b>			
	Format: MBZ			
	<b>Value</b>	<b>Name</b>	<b>Description</b>	
	0	Disable <b>[Default]</b>	This field must be programmed to 0	
<b>Programming Notes</b>				
This must be set to 0				

<b>SFC_STATE_BODY</b>		
11	<b>Reserved</b>	
	Access: RO	
	Format: MBZ	
	<b>10 Compression Type</b>	
	Format: U1	
	This field is applicable only when Memory compression is enabled.	
	<b>Value</b>	<b>Name</b>
	0	Media Compression Enabled <b>[Default]</b>
	1	Render Compression Enabled
	9	<b>Output Frame Surface Base Address - Memory Compression Enable</b>
Format: Enable		
Memory compression will be attempted for this surface.		
8:7	<b>Output Frame Surface Base Address - Arbitration Priority Control</b>	
	Format: HEVC_ARBITRATION_PRIORITY	
6:1	<b>Output Frame Surface Base Address - Index to Memory Object Control State (MOCS) Tables</b>	
	Format: U6	
	The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.	
	The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
0	<b>Reserved</b>	
19	<b>31:12 AVS Line Buffer Surface Base Address</b>	
	Specifies the 4K byte aligned frame buffer address for scratch space used for row/column store. This surface is used only if the internal buffer inside the SFC HW is not large enough to contain all row/column memory accesses. The AVS line buffer needs to be a valid address even for 1:1 scaling if SFC is used.	
	<b>Programming Notes</b>	
		This field needs to be programmed separately and exclusively for each pipe when SFC engine mode is not programmed 2'b00(Scalability workloads).
11:0	<b>Reserved</b>	
	Access: RO	
	Format: MBZ	
20	<b>31:16 Reserved</b>	
	Access: RO	
	Format: MBZ	

<b>SFC_STATE_BODY</b>																			
	15:0	<p><b>AVS Line Buffer Surface Base Address High</b></p> <p>This field is for the upper range [47:32] of AVS Line Buffer Surface Base Address. AVS Line buffer address needs to be valid even for 1:1 scaling if SFC is used.</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="3"><b>Programming Notes</b></td> </tr> <tr> <td colspan="3">This field needs to be programmed separately and exclusively for each pipe when SFC engine mode is not programmed 2'b00.</td> </tr> </table>	<b>Programming Notes</b>			This field needs to be programmed separately and exclusively for each pipe when SFC engine mode is not programmed 2'b00.													
	<b>Programming Notes</b>																		
This field needs to be programmed separately and exclusively for each pipe when SFC engine mode is not programmed 2'b00.																			
21	31:15	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ													
		Access:	RO																
	Format:	MBZ																	
	14:13	<p><b>AVS Line Buffer Tiled Mode</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2</td> </tr> </table> <p><b>For Media Surfaces:</b> This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
		Format:	U2																
		Value	Name	Description															
		0h	TRMODE_NONE	No tiled resource															
		1h	TRMODE_TILEYF	4KB tiled resources															
		2h	TRMODE_TILEYS	64KB tiled resources															
	3h	Reserved																	
12	<p><b>AVS Line Buffer Base Address - Row Store Scratch Buffer Cache Select</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U1</td> </tr> </table> <p>This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LLC</td> <td>Buffer going to LLC</td> </tr> </tbody> </table> <table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="3"><b>Programming Notes</b></td> </tr> <tr> <td colspan="3">This surface does not support to put in Row Store Scratch Buffer. Must be set to 0</td> </tr> </table>	Format:	U1	Value	Name	Description	0	LLC	Buffer going to LLC	<b>Programming Notes</b>			This surface does not support to put in Row Store Scratch Buffer. Must be set to 0						
	Format:	U1																	
	Value	Name	Description																
	0	LLC	Buffer going to LLC																
<b>Programming Notes</b>																			
This surface does not support to put in Row Store Scratch Buffer. Must be set to 0																			
11	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ														
	Access:	RO																	
Format:	MBZ																		
10	<p><b>AVS Line Buffer Base Address - Memory Compression Mode</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Default Value:</td> <td>0 Horizontal Compression Mode</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Distinguishes vertical from horizontal compression. Please refer to vol1a? <b>Memory Data Formats chapter - section</b> ?media Memory Compression for more details.</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="2"><b>Programming Notes</b></td> </tr> <tr> <td colspan="2">Memory compression is not supported. ?This bit is not used. Default to 0</td> </tr> </table>	Default Value:	0 Horizontal Compression Mode	Format:	U1	<b>Programming Notes</b>		Memory compression is not supported. ?This bit is not used. Default to 0											
	Default Value:	0 Horizontal Compression Mode																	
	Format:	U1																	
	<b>Programming Notes</b>																		
Memory compression is not supported. ?This bit is not used. Default to 0																			



<b>SFC_STATE_BODY</b>	
9	<b>AVS Line Buffer Base Address - Memory Compression Enable</b>
	Default Value: 0 Disable
	Format: Enable
	This bit control memory compression for this surface <div style="text-align: center;"><b>Programming Notes</b></div> This bit must be set to 0 (Memory compression is not supported in this surface)
8:7	<b>AVS Line Buffer Base Address - Arbitration Priority Control</b>
	Format: <b>HEVC_ARBITRATION_PRIORITY</b>
6:1	<b>AVS Line Buffer Base Address - Index to Memory Object Control State (MOCS) Tables</b>
	Format: U6
	The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.
	The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.
0	<b>Reserved</b>
22	31:12 <b>IEF Line Buffer Surface Base Address</b>
	Specifies the 4K byte aligned frame buffer address for the scratch space used for row/column store. This surface is used only if the internal buffer inside SFC HW is not large enough to contain all row/column memory accesses.
	<b>Programming Notes</b>
	This field needs to be programmed separately and exclusively for each pipe when SFC engine mode is not programmed 2'b00.
	<b>Restriction</b>
	For Integral Image Mode, this field is Reserved and MBZ.
11:0	<b>Reserved</b>
	Access: RO
	Format: MBZ
23	31:16 <b>Reserved</b>
	Access: RO
	Format: MBZ
	15:0 <b>IEF Line Buffer Surface Base Address High</b>
	This field is for the upper range [47:32] of IEF Line Buffer Surface Base Address. <div style="text-align: center;"><b>Programming Notes</b></div> This field needs to be programmed separately and exclusively for each pipe when SFC engine mode is not programmed 2'b00.

<b>SFC_STATE_BODY</b>																			
		<table border="1"> <tr> <th colspan="2" style="text-align: center;">Restriction</th> </tr> <tr> <td colspan="2">For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	Restriction		For Integral Image Mode, this field is Reserved and MBZ.														
Restriction																			
For Integral Image Mode, this field is Reserved and MBZ.																			
24	31:15	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ													
		Access:	RO																
	Format:	MBZ																	
	14:13	<b>IEF Line Buffer Tiled Mode</b> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p><b>For Media Surfaces:</b>This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
		Format:	U2																
		Value	Name	Description															
		0h	TRMODE_NONE	No tiled resource															
		1h	TRMODE_TILEYF	4KB tiled resources															
	2h	TRMODE_TILEYS	64KB tiled resources																
	3h	Reserved																	
12	<b>IEF Line Buffer Base Address - Row Store Scratch Buffer Cache Select</b> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LLC</td> <td>Buffer going to LLC</td> </tr> </tbody> </table> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0</td> </tr> </table>	Format:	U1	Value	Name	Description	0	LLC	Buffer going to LLC	Programming Notes		This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0							
Format:	U1																		
Value	Name	Description																	
0	LLC	Buffer going to LLC																	
Programming Notes																			
This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0																			
11	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ														
Access:	RO																		
Format:	MBZ																		
10	<b>IEF Line Buffer Base Address - Memory Compression Mode</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>Distinguishes vertical from horizontal compression.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Must be zero; memory compression is not supported for this surface. Default to 0</td> </tr> </table>	Default Value:	0	Format:	U1	Programming Notes		Must be zero; memory compression is not supported for this surface. Default to 0											
Default Value:	0																		
Format:	U1																		
Programming Notes																			
Must be zero; memory compression is not supported for this surface. Default to 0																			
9	<b>IEF Line Buffer Base Address - Memory Compression Enable</b> <table border="1"> <tr> <td>Default Value:</td> <td>0 Disable</td> </tr> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">Memory compression is not supported for this surface</td> </tr> </table>	Default Value:	0 Disable	Format:	Enable	Programming Notes		Memory compression is not supported for this surface											
Default Value:	0 Disable																		
Format:	Enable																		
Programming Notes																			
Memory compression is not supported for this surface																			

<b>SFC_STATE_BODY</b>		
		Must be 0.
	8:7	<b>IEF Line Buffer Base Address - Arbitration Priority Control</b> Format: <b>HEVC_ARBITRATION_PRIORITY</b>
	6:1	<b>IEF Line Buffer Base Address - Index to Memory Object Control State (MOCS) Tables</b> Format: U6  The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.  The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.
	0	<b>Reserved</b>
25	31:12	<b>SFD Line Buffer Surface Base Address</b> Specifies the 4K byte aligned frame buffer address for the scratch space used for row/column store. This surface is used only if the internal buffer inside SFC HW is not large enough to contain all row/column memory accesses.  <div style="text-align: center;"><b>Programming Notes</b></div> This field needs to be programmed separately and exclusively for each pipe when SFC engine mode is not programmed 2'b00.  <div style="text-align: center;"><b>Restriction</b></div> For Integral Image Mode, this field is Reserved and MBZ.
	11:0	<b>Reserved</b> Access: RO Format: MBZ
26	31:16	<b>Reserved</b> Access: RO Format: MBZ
	15:0	<b>SFD Line Buffer Surface Base Address High</b> This field is for the upper range [47:32] of SFD Line Buffer Surface Base Address.  <div style="text-align: center;"><b>Programming Notes</b></div> This field needs to be programmed separately and exclusively for each pipe when SFC engine mode is not programmed 2'b00.  <div style="text-align: center;"><b>Restriction</b></div> For Integral Image Mode, this field is Reserved and MBZ.
27	31:15	<b>Reserved</b> Access: RO Format: MBZ

## SFC\_STATE\_BODY

14:13	<b>SFD Line Buffer Tiled Mode</b>	Format:	U2
<b>For Media Surfaces:</b> This field specifies the tiled resource mode.			
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
		0h	TRMODE_NONE
		1h	TRMODE_TILEYF
		2h	TRMODE_TILEYS
		3h	Reserved
12	<b>SFD Line Buffer Base Address - Row Store Scratch Buffer Cache Select</b>	Format:	U1
This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.			
		<b>Value</b>	<b>Name</b>
		<b>Description</b>	
		0	LLC
		1	Media Storage <b>[Default]</b>
			Data will first cache in Media Storage
<b>Programming Notes</b>			
This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0			
11	<b>Reserved</b>	Access:	RO
		Format:	MBZ
10	<b>SFD Line Buffer Base Address - Memory Compression Mode</b>	Default Value:	0
		Format:	U1
Distinguishes vertical from horizontal compression. Please refer to vol1a <b>Memory Data Formats chapter - section</b> media Memory Compression for more details.			
<b>Programming Notes</b>			
Must be zero; memory compression is not supported for this surface. Default to0			
9	<b>SFD Line Buffer Base Address - Memory Compression Enable</b>	Default Value:	0 Disable
		Format:	Enable
<b>Programming Notes</b>			
Memory compression is not supported for this surface Must be 0.			
8:7	<b>SFD Line Buffer Base Address - Arbitration Priority Control</b>	Format:	<b>HEVC_ARBITRATION_PRIORITY</b>

## SFC\_STATE\_BODY

	6:1	<b>SFD Line Buffer Base Address - Index to Memory Object Control State (MOCS) Tables</b>				
		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	<b>Reserved</b>				
28	31:28	<b>Output Surface Format</b>				
	27	<b>Output Surface Interleave Chroma Enable</b>				
	26:22	<b>Reserved</b>				
		Access:	RO			
		Format:	MBZ			
	21:3	<b>Output Surface Pitch</b>				
		Format:	U19-1			
		This field specifies the surface pitch.				
		<b>Value</b>	<b>Name</b>	<b>Description</b>		
		[0,2047]	SURFTYPE_BUFFER Surfaces	[1B, 2048B]		
		[0, 524287]	Other Linear Surfaces	[64B, 512KB] = [1 CL, 8K CLs]		
		[511, 524287]	X-tiled Surface	[512B, 256KB] = [1tile, 512 tiles]		
		[127, 524287]	Y-tiled surfaces	[128B,256KB] = [1 tile, 2048 tiles]		
		<b>Programming Notes</b>				
		<ul style="list-style-type: none"> <li>For tiled surfaces, the pitch must be a multiple of the tile width</li> <li>For Linear surfaces, the pitch must be a multiple of CL (64B) width</li> <li>If <b>Half Pitch for Chroma</b> is set, this field must be a multiple of two tile widths for tiled surfaces, or a multiple of 2 bytes for linear surfaces.</li> </ul>				
		If Media Memory Compression is enabled, the following max pitch size restriction must be honored. For larger resolution, Media Memory compression Must be disabled.				
		Tiling Mode	Pixel Format	Max Frame Width (bytes)	Max Frame Width (pixels)	
		Legacy 4K	8bpp	16k	16k	
			16bpp	16k	8k	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

## SFC\_STATE\_BODY

SFC_STATE_BODY				
		32bpp	16k	4k
		64bpp	16k	2k
		128bpp	16k	1k
	TileYS	8bpp	16k	16k
		16bpp	16k	8k
		32bpp	16k	4k
		64bpp	16k	2k
		128bpp	16k	1k
2	<b>Output Surface Half Pitch For Chroma</b>			
	Exists If:	//PLANAR Surface Formats Only		
	Format:	Enable		
	This field indicates that the chroma plane(s) will use a pitch equal to half the value specified in the Surface Pitch field.			
1	<b>Output Surface Tiled</b>			
	Format:	Boolean		
	This field specifies whether the surface is tiled.			
	<b>Value</b>	<b>Name</b>	<b>Description</b>	
	1	True	Tiled	
	0	FALSE	Linear	
	<b>Programming Notes</b>			
	<ul style="list-style-type: none"> <li>Linear surfaces can be mapped to Main Memory (uncached) or System Memory (cacheable, snooped). Tiled surfaces can only be mapped to Main Memory.</li> <li>The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this bit.</li> </ul>			
0	<b>Output Surface Tile Walk</b>			
	Format:	Boolean		
	This field specifies the type of memory tiling (XMajor or YMajor) employed to tile this surface. See <i>Memory Interface Functions</i> for details on memory tiling and restrictions.			
	<b>Value</b>	<b>Name</b>		
	0	TILEWALK_XMAJOR		
	1	TILEWALK_YMAJOR		
	<b>Programming Notes</b>			
	<ul style="list-style-type: none"> <li>The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this bit.</li> </ul>			

<b>SFC_STATE_BODY</b>										
		This field is ignored when the surface is linear.								
29	31:16	<b>Output Surface X Offset For U</b> <table border="1"> <tr> <td>Exists If:</td> <td>//PLANAR Surface Formats Only</td> </tr> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>This field specifies the horizontal offset in pixels from the <b>Surface Base Address</b> to the start (origin) of the U(Cb) plane or the interleaved UV plane if <b>Interleave Chroma</b> is enabled.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For PLANAR_420 and PLANAR_422 surface formats, this field must be zero.</td> </tr> </table>	Exists If:	//PLANAR Surface Formats Only	Format:	U16	Programming Notes		For PLANAR_420 and PLANAR_422 surface formats, this field must be zero.	
		Exists If:	//PLANAR Surface Formats Only							
Format:	U16									
Programming Notes										
For PLANAR_420 and PLANAR_422 surface formats, this field must be zero.										
15:0	<b>Output Surface Y Offset For U</b> <table border="1"> <tr> <td>Exists If:</td> <td>//PLANAR Surface Formats Only</td> </tr> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>This field specifies the vertical offset in rows from the <b>Surface Base Address</b> to the start (origin) of the U(Cb) plane or the interleaved UV plane if <b>Interleave Chroma</b> is enabled.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For PLANAR_420 and PLANAR_422 surface formats, this field must be multiple of 16 pixels - i.e. multiple MBs.</td> </tr> </table>	Exists If:	//PLANAR Surface Formats Only	Format:	U16	Programming Notes		For PLANAR_420 and PLANAR_422 surface formats, this field must be multiple of 16 pixels - i.e. multiple MBs.		
		Exists If:	//PLANAR Surface Formats Only							
Format:	U16									
Programming Notes										
For PLANAR_420 and PLANAR_422 surface formats, this field must be multiple of 16 pixels - i.e. multiple MBs.										
30	31:16	<b>Output Surface X Offset For V</b> <table border="1"> <tr> <td>Exists If:</td> <td>//PLANAR Surface Formats with Interleaved Chroma Disable</td> </tr> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>This field specifies the horizontal offset in pixels from the <b>Surface Base Address</b> to the start (origin) of the V(Cr) plane.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.</td> </tr> </table>	Exists If:	//PLANAR Surface Formats with Interleaved Chroma Disable	Format:	U16	Programming Notes		For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.	
		Exists If:	//PLANAR Surface Formats with Interleaved Chroma Disable							
Format:	U16									
Programming Notes										
For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.										
15:0	<b>Output Surface Y Offset For V</b> <table border="1"> <tr> <td>Exists If:</td> <td>//PLANAR Surface Formats with Interleaved Chroma Disable</td> </tr> <tr> <td>Format:</td> <td>U16</td> </tr> </table> <p>This field specifies the vertical offset in rows from the Surface Base Address to the start (origin) of the V(Cr) plane.</p> <table border="1"> <tr> <th colspan="2" style="text-align: center;">Programming Notes</th> </tr> <tr> <td colspan="2">For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.</td> </tr> </table>	Exists If:	//PLANAR Surface Formats with Interleaved Chroma Disable	Format:	U16	Programming Notes		For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.		
		Exists If:	//PLANAR Surface Formats with Interleaved Chroma Disable							
Format:	U16									
Programming Notes										
For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.										
31	31:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
		Access:	RO							
Format:	MBZ									
32	31:0	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
		Access:	RO							
Format:	MBZ									

		<b>SFC_STATE_BODY</b>	
33	31:30	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	29:16	<b>SourceEndX</b>	
	Format:	U14	
	Indicates the X-direction end location in the original input frame to SFC. For 420/422 this field should be in multiple of 2.		
	<b>Programming Notes</b>		
	This field is only programmed when SFC Pipe Mode is HCP-to-SFC This should be in sync with tile widthsize programmed inHCP_TILE_CODING command		
15:14	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
13:0	<b>SourceStartX</b>		
	Format:	U14	
	Indicates the X-direction start location in the original input frame to SFC. For 420/422 this field should be in multiple of 2.		
	<b>Programming Notes</b>		
	This field is only programmed when SFC Pipe Mode is HCP-to-SFC This should be in sync with tile width size programmed in HCP_TILE_CODING command		
34	31:30	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	29:16	<b>DestinationEndX</b>	
	Format:	U14	
	Indicates the X-direction end location in the output frame of SFC.		
	<b>Programming Notes</b>		
	This field is valid only in Scalability Mode. Please refer to SFC Programming Model to program this field.		
15:14	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
13:0	<b>DestinationStartX</b>		
	Format:	U14	
	Indicate the X-direction start location in the output frame of SFC.		
	<b>Programming Notes</b>		
	This field is valid only in Scalability Mode. Please refer to SFC Programming Model to program this field.		



<b>SFC_STATE_BODY</b>					
35	31:29	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
28:5	<b>Xphaseshift</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>s4.19</td> </tr> </table> <p>Xphaseshift would be programmed to do output centering in x-direction.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>This field allows user to program the horizontal address/coordinate of the center of scaling. For the valid programming where the scaling center is within the original image, the numerical/floatingvalue for the Xphaseshift would be <math>c_x \cdot (1/sf\_hor - 1)</math>. The <math>sf\_hor</math> in the above equation is the numerical/floating value of the horizontal scaling factor while <math>c_x</math> corresponds to the normalized horizontal coordinate of the scaling center (i.e., <math>0 \leq c_x \leq 0.5</math>). For example, if <math>(c_x, c_y) = (0, 0)</math>, the scaling center would be the legacy top-left mode while <math>(c_x, c_y) = (0.5, 0.5)</math> would be the center mode which corresponds to the default of many other display solutions.</p>	Format:	s4.19	<b>Programming Notes</b>	
Format:	s4.19				
<b>Programming Notes</b>					
4:0	<b>Reserved</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				
36	31:29	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
28:5	<b>Yphaseshift</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>s4.19</td> </tr> </table> <p>Yphaseshift would be programmed to do output centering in y-direction.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>This field allows user to program the verticaladdress/coordinate of the center of scaling. For the valid programming where the scaling center is within the original image, the numerical/floatingvalue for the Yphaseshift would be <math>c_y \cdot (1/sf\_ver - 1)</math>. The <math>sf\_ver</math> in the above equation is the numerical/floating value of the verticalscaling factor while <math>c_y</math> corresponds to the normalized verticalcoordinate of the scaling center (i.e., <math>0 \leq c_y \leq 0.5</math>). For example, if <math>(c_x, c_y) = (0, 0)</math>, the scaling center would be the legacy top-left mode while <math>(c_x, c_y) = (0.5, 0.5)</math> would be the center mode which corresponds to the default of many other display solutions.</p>	Format:	s4.19	<b>Programming Notes</b>	
Format:	s4.19				
<b>Programming Notes</b>					
4:0	<b>Reserved</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				

<b>SFC_STATE_BODY</b>																			
37	31:12	<b>AVS Line Tile Buffer Surface Base Address</b> Specifies the 4K byte aligned frame buffer address for the scratch space used for row/column store. This surface is used only if the internal buffer inside SFC HW is not large enough to contain all row/column memory accesses.																	
		<b>Restriction</b> For Integral Image Mode, this field is Reserved and MBZ.																	
	11:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ													
Access:	RO																		
Format:	MBZ																		
38	31:16	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ													
		Access:	RO																
	Format:	MBZ																	
15:0	<b>AVS Line Tile Buffer Surface Base Address High</b> This field is for the upper range [47:32] of AVS Line Tile Buffer Surface Base Address.																		
<b>Restriction</b> For Integral Image Mode, this field is Reserved and MBZ.																			
39	31:15	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ													
		Access:	RO																
	Format:	MBZ																	
	14:13	<b>AVS Line Tile Buffer Tiled Mode</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2</td> </tr> </table> <p><b>For Media Surfaces:</b> This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
		Format:	U2																
Value		Name	Description																
0h		TRMODE_NONE	No tiled resource																
1h	TRMODE_TILEYF	4KB tiled resources																	
2h	TRMODE_TILEYS	64KB tiled resources																	
3h	Reserved																		
12	<b>AVS Line Tile Buffer Base Address - Row Store Scratch Buffer Cache Select</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U1</td> </tr> </table> This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.	Format:	U1																
Format:	U1																		
<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LLC <b>[Default]</b></td> <td>Buffer going to LLC</td> </tr> </tbody> </table>		Value	Name	Description	0	LLC <b>[Default]</b>	Buffer going to LLC												
Value	Name	Description																	
0	LLC <b>[Default]</b>	Buffer going to LLC																	
<b>Programming Notes</b> This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0																			

<b>SFC_STATE_BODY</b>				
40	11	<b>Reserved</b>	Access: RO	Format: MBZ
	10	<b>AVS Line Tile Buffer Base Address - Memory Compression Mode</b>	Default Value: 0	Format: U1
	Distinguishes vertical from horizontal compression. Please refer to vol1a <b>Memory Data Formats chapter - section</b> media Memory Compression for more details.			
	<b>Programming Notes</b>			
	Must be zero; memory compression is not supported for this surface. Default to 0			
	9	<b>AVS Line Tile Buffer Base Address - Memory Compression Enable</b>	Default Value: 0 Disable	Format: Enable
	<b>Programming Notes</b>			
	Memory compression is not supported for this surface Must be 0.			
	8:7	<b>AVS Line Tile Buffer Base Address - Arbitration Priority Control</b>	Format: <b>HEVC_ARBITRATION_PRIORITY</b>	
	6:1	<b>AVS Line Tile Buffer Base Address - Index to Memory Object Control State (MOCS) Tables</b>	Format: U6	
The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.				
The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.				
0	<b>Reserved</b>			
31:12	<b>IEF Line Tile Buffer Surface Base Address</b>	Specifies the 4K byte aligned frame buffer address for the scratch space used for row/column store. This surface is used only if the internal buffer inside SFC HW is not large enough to contain all row/column memory accesses.		
<b>Restriction</b>				
For Integral Image Mode, this field is Reserved and MBZ.				
11:0	<b>Reserved</b>	Access: RO	Format: MBZ	

<b>SFC_STATE_BODY</b>																			
41	31:16	<b>Reserved</b>																	
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ													
	Access:	RO																	
	Format:	MBZ																	
15:0	<b>IEF Line Tile Buffer Surface Base Address High</b> This field is for the upper range [47:32] of IEF Line Tile Buffer Surface Base Address. <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <th style="text-align: center;">Restriction</th> </tr> <tr> <td>For Integral Image Mode, this field is Reserved and MBZ.</td> </tr> </table>	Restriction	For Integral Image Mode, this field is Reserved and MBZ.																
Restriction																			
For Integral Image Mode, this field is Reserved and MBZ.																			
31:15	<b>Reserved</b> <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ														
Access:	RO																		
Format:	MBZ																		
42	14:13	<b>IEF Line Tile Buffer Tiled Mode</b> <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 60%;">Format:</td> <td>U2</td> </tr> </table> <b>For Media Surfaces:</b> This field specifies the tiled resource mode. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
		Format:	U2																
	Value	Name	Description																
	0h	TRMODE_NONE	No tiled resource																
	1h	TRMODE_TILEYF	4KB tiled resources																
	2h	TRMODE_TILEYS	64KB tiled resources																
	3h	Reserved																	
	12	<b>IEF Line Tile Buffer Base Address - Row Store Scratch Buffer Cache Select</b> <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 60%;">Format:</td> <td>U1</td> </tr> </table> This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC. <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LLC <b>[Default]</b></td> <td>Buffer going to LLC</td> </tr> </tbody> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0</td> </tr> </table>	Format:	U1	Value	Name	Description	0	LLC <b>[Default]</b>	Buffer going to LLC	Programming Notes	This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0							
	Format:	U1																	
	Value	Name	Description																
0	LLC <b>[Default]</b>	Buffer going to LLC																	
Programming Notes																			
This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0																			
11	<b>Reserved</b> <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ														
	Access:	RO																	
Format:	MBZ																		
10	<b>IEF Line Tile Buffer Base Address - Memory Compression Mode</b> <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U1</td> </tr> </table> Distinguishes vertical from horizontal compression. Please refer to vol1a <b>Memory Data Formats chapter - section</b> media Memory Compression for more details. <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>Must be zero; memory compression is not supported for this surface. Default to 0</td> </tr> </table>	Default Value:	0	Format:	U1	Programming Notes	Must be zero; memory compression is not supported for this surface. Default to 0												
Default Value:	0																		
Format:	U1																		
Programming Notes																			
Must be zero; memory compression is not supported for this surface. Default to 0																			

<b>SFC_STATE_BODY</b>		
	9	<b>IEF Line Tile Buffer Base Address - Memory Compression Enable</b>
		Default Value: 0 Disable
		Format: Enable
		<p style="text-align: center;"><b>Programming Notes</b></p> <p>Memory compression is not supported for this surface Must be 0.</p>
	8:7	<b>IEF Line Tile Buffer Base Address - Arbitration Priority Control</b>
		Format: <b>HEVC_ARBITRATION_PRIORITY</b>
	6:1	<b>IEF Line Tile Buffer Base Address - Index to Memory Object Control State (MOCS) Tables</b>
		Format: U6
		The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.
		The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.
	0	<b>Reserved</b>
43	31:12	<b>SFD Line Tile Buffer Surface Base Address</b>
		Specifies the 4K byte aligned frame buffer address for the scratch space used for row/column store. This surface is used only if the internal buffer inside SFC HW is not large enough to contain all row/column memory accesses.
		<p style="text-align: center;"><b>Restriction</b></p> <p>For Integral Image Mode, this field is Reserved and MBZ.</p>
	11:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
44	31:16	<b>Reserved</b>
		Access: RO
		Format: MBZ
	15:0	<b>SFD Line Tile Buffer Surface Base Address High</b>
		This field is for the upper range [47:32] of SFD Line Tile Buffer Surface Base Address.
		<p style="text-align: center;"><b>Restriction</b></p> <p>For Integral Image Mode, this field is Reserved and MBZ.</p>
45	31:15	<b>Reserved</b>
		Access: RO
		Format: MBZ

## SFC\_STATE\_BODY

14:13	<b>SFD Line Tile Buffer Tiled Mode</b>	Format:	U2															
<p><b>For Media Surfaces:</b>This field specifies the tiled resource mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>				Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved	
Value	Name	Description																
0h	TRMODE_NONE	No tiled resource																
1h	TRMODE_TILEYF	4KB tiled resources																
2h	TRMODE_TILEYS	64KB tiled resources																
3h	Reserved																	
12	<b>SFD Line Tile Buffer Base Address - Row Store Scratch Buffer Cache Select</b>	Format:	U1															
<p>This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 35%;">Name</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LLC <b>[Default]</b></td> <td>Buffer going to LLC</td> </tr> </tbody> </table>				Value	Name	Description	0	LLC <b>[Default]</b>	Buffer going to LLC									
Value	Name	Description																
0	LLC <b>[Default]</b>	Buffer going to LLC																
<b>Programming Notes</b>																		
This surface does not support Rowstore Scratch Buffer Cache. Must be programmed to 0																		
11	<b>Reserved</b>	Access:	RO															
		Format:	MBZ															
10	<b>SFD Line Tile Buffer Base Address - Memory Compression Mode</b>	Default Value:	0															
		Format:	U1															
<p>Distinguishes vertical from horizontal compression. Please refer to vol1a <b>Memory Data Formats chapter - section</b> media Memory Compression for more details.</p>																		
<b>Programming Notes</b>																		
Must be zero; memory compression is not supported for this surface. Default to 0																		
9	<b>SFD Line Tile Buffer Base Address - Memory Compression Enable</b>	Default Value:	0 Disable															
		Format:	Enable															
<b>Programming Notes</b>																		
Memory compression is not supported for this surface Must be 0.																		
8:7	<b>SFD Line Tile Buffer Base Address - Arbitration Priority Control</b>	Format:	<b>HEVC_ARBITRATION_PRIORITY</b>															

<b>SFC_STATE_BODY</b>																		
	6:1	<p><b>SFD Line Tile Buffer Base Address - Index to Memory Object Control State (MOCS) Tables</b></p> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.</p> <p>The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.</p>	Format:	U6														
	Format:	U6																
	0	<b>Reserved</b>																
46	<p>31:12 <b>Histogram Surface Base Address</b> Specifies the CL aligned frame buffer address for Y histogram. The <b>SFC_HISTOGRAM_SURFACE</b> specifies how histogram data would be places in surface.</p> <p>11:0 <b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ													
Access:	RO																	
Format:	MBZ																	
47	<p>31:16 <b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table> <p>15:0 <b>Histogram Surface Base Address High</b> This field is for the upper range [47:32] of Histogram Surface Base Address.</p>	Access:	RO	Format:	MBZ													
Access:	RO																	
Format:	MBZ																	
48	31:15	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ												
	Access:	RO																
	Format:	MBZ																
	14:13	<p><b>Histogram Tiled Mode</b></p> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p><b>For Media Surfaces:</b>This field specifies the tiled resource mode.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>TRMODE_NONE</td> <td>No tiled resource</td> </tr> <tr> <td>1h</td> <td>TRMODE_TILEYF</td> <td>4KB tiled resources</td> </tr> <tr> <td>2h</td> <td>TRMODE_TILEYS</td> <td>64KB tiled resources</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	Format:	U2	Value	Name	Description	0h	TRMODE_NONE	No tiled resource	1h	TRMODE_TILEYF	4KB tiled resources	2h	TRMODE_TILEYS	64KB tiled resources	3h	Reserved
Format:	U2																	
Value	Name	Description																
0h	TRMODE_NONE	No tiled resource																
1h	TRMODE_TILEYF	4KB tiled resources																
2h	TRMODE_TILEYS	64KB tiled resources																
3h	Reserved																	
12	<p><b>Histogram Base Address - Cache Select</b></p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>This field controls if the Histogram need to be cached in LLC or not.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LLC <b>[Default]</b></td> <td>Buffer going to LLC</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>Must be programmed to 0</p>	Format:	U1	Value	Name	Description	0	LLC <b>[Default]</b>	Buffer going to LLC									
Format:	U1																	
Value	Name	Description																
0	LLC <b>[Default]</b>	Buffer going to LLC																

<b>SFC_STATE_BODY</b>		
11	<b>Reserved</b>	
	Access:	RO
	Format:	MBZ
10	<b>Histogram Base Address - Memory Compression Type</b>	
	Default Value:	0
	Format:	U1
	Distinguishes Media or 3D compression. <b>Memory Data Formats chapter - section</b> media Memory Compression for more details.	
	<b>Programming Notes</b>	
Must be zero; memory compression is not supported for this surface. Default to 0		
9	<b>Histogram Base Address - Memory Compression Enable</b>	
	Default Value:	0 Disable
	Format:	Enable
	<b>Programming Notes</b>	
	Memory compression is not supported for this surface Must be 0.	
8:7	<b>Histogram Base Address - Arbitration Priority Control</b>	
	Format:	<b>HEVC_ARBITRATION_PRIORITY</b>
6:1	<b>Histogram Base Address - Index to Memory Object Control State (MOCS) Tables</b>	
	Format:	U6
	The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.	
	The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.	
0	<b>Reserved</b>	



## SIMD1 Untyped BUFFER Surface 64-Bit Address Payload

MAP64B_USU_SIMD1 - SIMD1 Untyped BUFFER Surface 64-Bit Address Payload				
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.1	63:0	<p><b>U0</b></p> <table border="1"> <tr> <td>Format:</td> <td>U64</td> </tr> </table> <p>Specifies the U channel for slot [0]</p>	Format:	U64
Format:	U64			



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



## SIMD8 LOD Message Address Payload Control

<b>MACR_LOD_SIMD8 - SIMD8 LOD Message Address Payload Control</b>		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0	31:0	<b>Slot0 LOD</b> Format: <span style="border: 1px solid black; padding: 2px;">MACD_LOD</span> Specifies the LOD for slot 0
0.1	31:0	<b>Slot1 LOD</b> Format: <span style="border: 1px solid black; padding: 2px;">MACD_LOD</span> Specifies the LOD for slot 1
0.2	31:0	<b>Slot2 LOD</b> Format: <span style="border: 1px solid black; padding: 2px;">MACD_LOD</span> Specifies the LOD for slot 2
0.3	31:0	<b>Slot3 LOD</b> Format: <span style="border: 1px solid black; padding: 2px;">MACD_LOD</span> Specifies the LOD for slot 3
0.4	31:0	<b>Slot4 LOD</b> Format: <span style="border: 1px solid black; padding: 2px;">MACD_LOD</span> Specifies the LOD for slot 4
0.5	31:0	<b>Slot5 LOD</b> Format: <span style="border: 1px solid black; padding: 2px;">MACD_LOD</span> Specifies the LOD for slot 5
0.6	31:0	<b>Slot6 LOD</b> Format: <span style="border: 1px solid black; padding: 2px;">MACD_LOD</span> Specifies the LOD for slot 6
0.7	31:0	<b>Slot7 LOD</b> Format: <span style="border: 1px solid black; padding: 2px;">MACD_LOD</span> Specifies the LOD for slot 7

## SIMD8 Render Target Data Payload

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



## SIMD8 Typed Surface 32-Bit Address Payload

<b>MAP32B_TS_SIMD8 - SIMD8 Typed Surface 32-Bit Address Payload</b>		
Size (in bits):	1024	
Default Value:	0x00000000, 0x00000000	
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0.0-0.7	255:0	<b>U</b> Format: <b>MACR_32b</b> Specifies the U channel for slots [7:0]
1.0-1.7	255:0	<b>V</b> Format: <b>MACR_32b</b> Specifies the V channel for slots [7:0]
2.0-2.7	255:0	<b>R</b> Format: <b>MACR_32b</b> Specifies the R channel for slots [7:0]
3.0-3.7	255:0	<b>LOD</b> Format: <b>MACR_LOD_SIMD8</b> Specifies the LOD for slots [7:0]

## SIMD8 Untyped BUFFER Surface 32-Bit Address Payload

MAP32B_USU_SIMD8 - SIMD8 Untyped BUFFER Surface 32-Bit Address Payload		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>U</b> Format: <b>MACR_32b</b> Specifies the U channel for slots [7:0]



## SIMD8 Untyped BUFFER Surface 64-Bit Address Payload

MAP64B_USU_SIMD8 - SIMD8 Untyped BUFFER Surface 64-Bit Address Payload		
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	<b>U3_U0</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MACR_64b</b></td> </tr> </table> <p>Specifies the U channel for slots [3:0]</p>
Format:	<b>MACR_64b</b>	
1.0-1.7	255:0	<b>U7_U4</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MACR_64b</b></td> </tr> </table> <p>Specifies the U channel for slots [7:4]</p>
Format:	<b>MACR_64b</b>	



## SIMD8 Untyped SCRATCH Surface 32-Bit Address Payload

MAP32B_USUV_SIMD8 - SIMD8 Untyped SCRATCH Surface 32-Bit Address Payload		
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	<b>U</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MACR_32b</b></td> </tr> </table> <p>Specifies the U channel for slots [7:0]</p>
Format:	<b>MACR_32b</b>	
1.0-1.7	255:0	<b>V</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MACR_32b</b></td> </tr> </table> <p>Specifies the V channel for slots [7:0]</p>
Format:	<b>MACR_32b</b>	



## SIMD8 URB Channel Mask Message Address Payload

<b>MAPU_CMASK_SIMD8 - SIMD8 URB Channel Mask Message Address Payload</b>		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Per Slot Channel Mask</b> Format: <b>MACD_URB_CMASK[8]</b> Each slot's mask field is combined with the execution mask to determine which Dwords are written to the URB.

## SIMD8 URB Offset Message Address Payload

MAPU_SIMD8 - SIMD8 URB Offset Message Address Payload								
Size (in bits):	256							
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000							
DWord	Bit	Description						
0.0-0.7	255:0	<p><b>Slot Offset</b></p> <table border="1"> <tr> <td>Format:</td> <td>U32[8]</td> </tr> </table> <p>Each slot's offset field is added to the <b>Global Offset</b>(specified in the message descriptor) and the slot's URB Handle (specified in the message header)to generate the URB address for this access. This offset and the Global Offset are specified as Oword units (128 bits).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0-2047]</td> <td></td> </tr> </tbody> </table>	Format:	U32[8]	Value	Name	[0-2047]	
Format:	U32[8]							
Value	Name							
[0-2047]								



<b>MDP_RTW_16 - SIMD16 Render Target Data Payload</b>			
		Format:	<b>MDP_DW_SIMD8</b>
		Slots [7:0] Alpha	
7.0-7.7	255:0	<b>Alpha[15:7]</b>	
		Format:	<b>MDP_DW_SIMD8</b>
		Slots [15:7] Alpha	



## SIMD16 Untyped BUFFER Surface 16-Bit Address Payload

MAP16B_USU_SIMD16 - SIMD16 Untyped BUFFER Surface 16-Bit Address Payload				
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<b>U</b> <table border="1" data-bbox="560 657 1469 703"> <tr> <td>Format:</td> <td>U16[16]</td> </tr> </table> Specifies the U channel for slots [15:0]	Format:	U16[16]
Format:	U16[16]			



## SIMD16 Untyped BUFFER Surface 32-Bit Address Payload

MAP32B_USU_SIMD16 - SIMD16 Untyped BUFFER Surface 32-Bit Address Payload				
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		
0.0-1.7	511:0	<b>U</b> <table border="1"> <tr> <td>Format:</td> <td>U32[16]</td> </tr> </table> Specifies the U channel for slots [15:0]	Format:	U32[16]
Format:	U32[16]			







## SIMD16 Untyped SCRATCH Surface 16-Bit Address Payload

MAP16B_USUV_SIMD16 - SIMD16 Untyped SCRATCH Surface 16-Bit Address Payload		
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	<b>U</b>
		<table border="1"> <tr> <td>Format:</td> <td>U16[16]</td> </tr> </table> <p>Specifies the U channel for slots [15:0]</p>
Format:	U16[16]	
1.0-1.7	255:0	<b>V</b>
		<table border="1"> <tr> <td>Format:</td> <td>U16[16]</td> </tr> </table> <p>Specifies the V channel for slots [15:0]</p>
Format:	U16[16]	

## SIMD16 Untyped SCRATCH Surface 32-Bit Address Payload

<b>MAP32B_USUV_SIMD16 - SIMD16 Untyped SCRATCH Surface 32-Bit Address Payload</b>				
Size (in bits):	1024			
Default Value:	0x00000000, 0x00000000			
<b>DWord</b>	<b>Bit</b>	<b>Description</b>		
0.0-1.7	511:0	<b>U</b> <table border="1"> <tr> <td>Format:</td> <td>U32[16]</td> </tr> </table> Specifies the U channel for slots [15:0]	Format:	U32[16]
Format:	U32[16]			
2.0-3.7	511:0	<b>V</b> <table border="1"> <tr> <td>Format:</td> <td>U32[16]</td> </tr> </table> Specifies the V channel for slots [15:0]	Format:	U32[16]
Format:	U32[16]			



## SIMD 32-Bit Address Payload Control

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

## SIMD 64-Bit Address Payload Control

MACR_64B - SIMD 64-Bit Address Payload Control		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.1	63:0	<b>Offset0</b>
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U64</td></tr></table> Specifies the address offset for slot 0 in this payload register.
	U64	
0.2-0.3	63:0	<b>Offset1</b>
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U64</td></tr></table> Specifies the address offset for slot 1 in this payload register.
	U64	
0.4-0.5	63:0	<b>Offset2</b>
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U64</td></tr></table> Specifies the address offset for slot 2 in this payload register.
	U64	
0.6-0.7	63:0	<b>Offset3</b>
		Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U64</td></tr></table> Specifies the address offset for slot 3 in this payload register.
	U64	



## SIMD8 32-Bit Address Payload

MAP32B_SIMD8 - SIMD8 32-Bit Address Payload		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Offset[7:0]</b> Format: <b>MACR_32b</b> Specifies the address offset for Slots [7:0].

## SIMD8 64-Bit Address Payload

MAP64B_SIMD8 - SIMD8 64-Bit Address Payload		
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	<b>Offset[3:0]</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MACR_64b</b></td> </tr> </table> <p>Specifies the address offset for slots [3:0].</p>
Format:	<b>MACR_64b</b>	
1.0-1.7	255:0	<b>Offset[7:4]</b>
		<table border="1"> <tr> <td>Format:</td> <td><b>MACR_64b</b></td> </tr> </table> <p>Specifies the address offset for slots [7:4].</p>
Format:	<b>MACR_64b</b>	



## SIMD16 16-Bit Address Payload

MAP16B_SIMD16 - SIMD16 16-Bit Address Payload		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Offset</b> Format: U16[16] Specifies the address offset for slots [15:0].



## SIMD16 32-Bit Address Payload

MAP32B_SIMD16 - SIMD16 32-Bit Address Payload				
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		
0.0-1.7	511:0	<b>Offset</b> <table border="1" data-bbox="544 640 1469 682"> <tr> <td>Format:</td> <td>U32[16]</td> </tr> </table> Specifies the address offset for slots [15:0].	Format:	U32[16]
Format:	U32[16]			



## SIMD16 64-Bit Address Payload

MAP64B_SIMD16 - SIMD16 64-Bit Address Payload		
Size (in bits):	1024	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Offset[3:0]</b> Format: <b>MACR_64b</b> Specifies the address offsets for slots [3:0].
1.0-1.7	255:0	<b>Offset[7:4]</b> Format: <b>MACR_64b</b> Specifies the address offsets for slots [7:4].
2.0-2.7	255:0	<b>Offset[11:8]</b> Format: <b>MACR_64b</b> Specifies the address offsets for slots [11:8].
3.0-3.7	255:0	<b>Offset[15:12]</b> Format: <b>MACR_64b</b> Specifies the address offsets for slots [15:12].

## SIMD Mode 2 Message Descriptor Control Field

MDC_SM2 - SIMD Mode 2 Message Descriptor Control Field											
Size (in bits):		1									
Default Value:		0x00000000									
DWord	Bit	Description									
0	0	<p><b>SIMD Mode</b> Specifies the SIMD mode of the message (number of slots processed)</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>SIMD8</td> <td>SIMD8</td> </tr> <tr> <td>01h</td> <td>SIMD16</td> <td>SIMD16</td> </tr> </tbody> </table>	Value	Name	Description	00h	SIMD8	SIMD8	01h	SIMD16	SIMD16
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																	
Size (in bits):	2																
Default Value:	0x00000000																
DWord	Bit	Description															
0	1:0	<b>SIMD Mode</b> 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>Reserved</td><td>Ignored</td></tr><tr><td>01h</td><td>SIMD16</td><td>SIMD16</td></tr><tr><td>02h</td><td>SIMD8</td><td>SIMD8</td></tr><tr><td>03h</td><td>Reserved</td><td>Ignored</td></tr></tbody></table>	Value	Name	Description	00h	Reserved	Ignored	01h	SIMD16	SIMD16	02h	SIMD8	SIMD8	03h	Reserved	Ignored
Value	Name	Description															
00h	Reserved	Ignored															
01h	SIMD16	SIMD16															
02h	SIMD8	SIMD8															
03h	Reserved	Ignored															

## SLICE\_HASH\_TABLE

SLICE_HASH_TABLE - SLICE_HASH_TABLE		
Size (in bits):	1024	
Default Value:	0x00000000, 0x00000000	
Description		
<p>The slice hash table state is stored as an array tables (2 slices-8 slices), each of which contains the 32 DWords described here. 16x16 lookup table for slice indexed by lower bits of pixel block address. Each entry in the table indicates the <b>physical</b> slice_id to map that XY. If a slice is disabled, then it must not be present in the table. Entries in the table that point to disabled slice will be mapped to lowest enabled slice_id.</p>		
DWord	Bit	Description
0..31	1023:0	<b>Slice Hashing Table Entries</b> Format: <b>SLICE_HASHING_TABLE_ENTRY[16]</b> Each entry has the Enabled Physical Iz PipeID to map that [Y][X] pixel block address.



## SLICE\_HASHING\_TABLE\_ENTRY

SLICE_HASHING_TABLE_ENTRY		
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0	31:28	<b>Entry7</b> Format: <input type="text"/> U4
	27:24	<b>Entry6</b> Format: <input type="text"/> U4
	23:20	<b>Entry5</b> Format: <input type="text"/> U4
	19:16	<b>Entry4</b> Format: <input type="text"/> U4
	15:12	<b>Entry3</b> Format: <input type="text"/> U4
	11:8	<b>Entry2</b> Format: <input type="text"/> U4
	7:4	<b>Entry1</b> Format: <input type="text"/> U4
	3:0	<b>Entry0</b> Format: <input type="text"/> U4
1	31:28	<b>Entry15</b>
	27:24	<b>Entry14</b> Format: <input type="text"/> U4
	23:20	<b>Entry13</b> Format: <input type="text"/> U4
	19:16	<b>Entry12</b> Format: <input type="text"/> U4
	15:12	<b>Entry11</b> Format: <input type="text"/> U4
	11:8	<b>Entry10</b> Format: <input type="text"/> U4
	7:4	<b>Entry9</b> Format: <input type="text"/> U4
	3:0	<b>Entry8</b> Format: <input type="text"/> U4

## SLM Block Message Header

<b>MH_SLM_GO - SLM Block Message Header</b>		
Source:	EuSubFunctionDataPort0	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
2	31:0	<b>Global Offset</b>
		Format: U32
		Specifies the global element index into the buffer, in units of Hwords, Owords, Dwords, or Bytes (depending on the message).
		<b>Programming Notes</b>
		The Global Offset for Aligned Block operations is specified as a Dword-aligned byte offset (offset bits [1:0] = 0), or Oword-aligned byte offset (offset bits [3:0]=0), or Hword-aligned byte offset (offset bits [4:0]=0).
3..7	159:0	<b>Reserved</b>
		Access: RO
		Format: MBZ



## Slot Group 2 Message Descriptor Control Field

MDC_SG2 - Slot Group 2 Message Descriptor Control Field											
Size (in bits):		1									
Default Value:		0x00000000									
DWord	Bit	Description									
0	0	<b>SIMD Mode</b> 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.									
		<table border="1"><thead><tr><th>Value</th><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>00h</td><td>SG8L</td><td>Use low 8 slots</td></tr><tr><td>01h</td><td>SG8U</td><td>Use high 8 slots</td></tr></tbody></table>	Value	Name	Description	00h	SG8L	Use low 8 slots	01h	SG8U	Use high 8 slots
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																	
Size (in bits):		2															
Default Value:		0x00000000															
DWord	Bit	Description															
0	1:0	<b>SIMD Mode</b> 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.															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Reserved</td> <td>Ignored</td> </tr> <tr> <td>01h</td> <td>SG8L</td> <td>[] Use low 8 slots</td> </tr> <tr> <td>02h</td> <td>SG8U</td> <td>Use high 8 slots</td> </tr> <tr> <td>03h</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Value	Name	Description	00h	Reserved	Ignored	01h	SG8L	[] Use low 8 slots	02h	SG8U	Use high 8 slots	03h	Reserved	Ignored
		Value	Name	Description													
		00h	Reserved	Ignored													
		01h	SG8L	[] Use low 8 slots													
02h	SG8U	Use high 8 slots															
03h	Reserved	Ignored															
00h	Reserved	Ignored															
01h	SG8L	[] Use low 8 slots															
02h	SG8U	Use high 8 slots															
03h	Reserved	Ignored															

## Slot Group Select Render Cache Message Descriptor Control Field

<b>MDC_RT_SGS - Slot Group Select Render Cache Message Descriptor Control Field</b>											
Size (in bits):		1									
Default Value:		0x00000000									
DWord	Bit	Description									
0	0	<p><b>Slot Group Select</b></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

<b>SO_DECL</b>								
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	<b>Reserved</b>						
		<table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
	Access:	RO						
	Format:	MBZ						
	13:12	<b>Output Buffer Slot</b>						
<table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>This field selects the destination output buffer slot.</p>		Format:	U2					
Format:	U2							
11	11	<b>Hole Flag</b>						
		<table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table> <p>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:</p> <table border="1" style="width: 100%;"> <tr> <td>0x0 No Dwords are skipped over (SO_DECL performs no operation)</td> </tr> <tr> <td>0x1 (X) Skip 1 DWord</td> </tr> <tr> <td>0x3 (XY) Skip 2 DWords</td> </tr> <tr> <td>0x7 (XYZ) Skip 3 DWords</td> </tr> <tr> <td>0xF (XYZW) Skip 4 DWords</td> </tr> </table>	Format:	Enable	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
	Format:	Enable						
	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	<b>Reserved</b>							
	<table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
Access:	RO							
Format:	MBZ							
9:4	<b>Register Index</b>							
	<table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>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)</p> <p>There is only enough internal storage for the 128-bit vertex header and 32 128-bit vertex</p>	Format:	U6					
Format:	U6							

SO_DECL															
	<p>attributes.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Value</th> <th style="width: 50%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,32]</td> <td></td> </tr> <tr> <td>0h</td> <td><b>[Default]</b></td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>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.</p>	Value	Name	[0,32]		0h	<b>[Default]</b>								
Value	Name														
[0,32]															
0h	<b>[Default]</b>														
3:0	<p><b>Component Mask</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U4</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 <b>Hole Flag</b> is set, this field (if non-zero) indirectly specifies how much the destination buffer's write pointer should be advanced. See <b>Hole Flag</b> description above for restrictions on this field. If the <b>Hole Flag</b> 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: 20%;">Value</th> <th style="width: 80%;">Name</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>SO_DECL_COMPMASK_NONE <b>[Default]</b></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:	U4	Value	Name	0h	SO_DECL_COMPMASK_NONE <b>[Default]</b>	xxx1b	SO_DECL_COMPMASK_X	xx1xb	SO_DECL_COMPMASK_Y	x1xxb	SO_DECL_COMPMASK_Z	1xxxb	SO_DECL_COMPMASK_W
Format:	U4														
Value	Name														
0h	SO_DECL_COMPMASK_NONE <b>[Default]</b>														
xxx1b	SO_DECL_COMPMASK_X														
xx1xb	SO_DECL_COMPMASK_Y														
x1xxb	SO_DECL_COMPMASK_Z														
1xxxb	SO_DECL_COMPMASK_W														

## SO\_DECL\_ENTRY

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

## Split\_coding\_unit\_flags

<b>Split_coding_unit_flags</b>					
Source:	VideoCS				
Size (in bits):	21				
Default Value:	0x00000000				
Contains the split level flags, level 0 through 2.					
DWord	Bit	Description			
0	20	<b>Split_flag_level0</b>			
		Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U1</td></tr></table>		U1	
		U1			
	19:16	<b>Split_flag_level1</b>	Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U4</td></tr></table>		U4
				U4	
			[19:16] is in raster order. Bit16 is for partition0 in raster order.		
	15:12	<b>Split_flag_level2 level1part3</b>	Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U4</td></tr></table>		U4
				U4	
			Split flags for bit19 partition.		
			[15:12] is in raster order. Bit12 is for partition0 in raster order.		
11:8	<b>Split_flag_level2 level1part2</b>	Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U4</td></tr></table>		U4	
			U4		
		Split flags for bit18 partition.			
[11:8] is in raster order. Bit8 is for partition0 in raster order.					
7:4	<b>Split_flag_level2 level1part1</b>	Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U4</td></tr></table>		U4	
			U4		
		Split flags for bit17 partition.			
[7:4] is in raster order. Bit4 is for partition0 in raster order.					
3:0	<b>Split_flag_level2 level1part0</b>	Format: <table border="1" style="width: 100%;"><tr><td style="width: 80%;"></td><td style="width: 20%; text-align: center;">U4</td></tr></table>		U4	
			U4		
		Split flags for bit16 partition.			
[3:0] is in raster order. Bit0 is for partition0 in raster order.					

## SplitBaseAddress4KByteAligned

<b>SplitBaseAddress4KByteAligned</b>		
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
Specifies a 64-bit (48-bit canonical) 4K-byte aligned memory base address. GraphicsAddress is a 64-bit value [63:0], but only a portion of it is used by hardware. The upper reserved bits are ignored and MBZ.		
<b>Programming Notes</b>		
Bits 63:48 must be zero.		
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0..1	63:12	<b>Base Address</b>
		Format: VIRTUAL_ADDR[63:12]
	11:0	<b>Reserved</b>
		Access: RO
		Format: MBZ



## SplitBaseAddress64ByteAligned

<b>SplitBaseAddress64ByteAligned</b>		
Size (in bits):	64	
Default Value:	0x00000000, 0x00000000	
Specifies a 64-bit (48-bit canonical) 64-byte aligned memory base address.		
<b>Programming Notes</b>		
Bits 63:48 must be zero.		
DWord	Bit	Description
0..1	63:6	<b>Base Address</b>
		Format: VIRTUAL_ADDR[63:6]
	5:0	<b>Reserved</b>
		Access: RO Format: MBZ



## Stateless Binding Table Index Message Descriptor Control Field

<b>MDC_STATELESS - Stateless Binding Table Index Message Descriptor Control Field</b>											
Size (in bits):		8									
Default Value:		0x00000000									
DWord	Bit	Description									
0	7:0	<b>Binding Table Index</b> Specifies the message is Stateless									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <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	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							
		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

<b>MH_A32_GO - Stateless Block Message Header</b>		
Source:	EuSubFunctionDataPort0	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..1	63:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
2	31:0	<b>Global Offset</b>
		Format: U32
		Specifies the global element index into the buffer, in units of Owords, Dwords, or Bytes (depending on the message).
		<b>Programming Notes</b>
		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	<b>Per Thread Scratch Space</b>
		Format: <b>MHC_PTSS</b>
		Specifies amount of scratch space used by this thread, for Stateless bounds checking.
4	31:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
5	31:0	<b>Buffer Base Address</b>
		Format: <b>MHC_A32_BBA</b>
		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[31:10] must be less than $2^{48}$ . It is illegal for this to be greater or equal than $2^{48}$ .
6..7	63:0	<b>Reserved</b>
		Access: RO
		Format: MBZ

## Stateless Surface Message Header

<b>MH1_A32 - Stateless Surface Message Header</b>		
Source:	EuSubFunctionDataPort1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..4	159:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
5	31:0	<b>Buffer Base Address</b>
		Format: <b>MHC_A32_BBA</b> Specifies the surface address offset page [31:10] for A32 stateless messages.
6..7	63:0	<b>Reserved</b>
		Access: RO
		Format: MBZ



## Stateless Surface Pixel Mask Message Header

MH1_A32_PSM - Stateless Surface Pixel Mask Message Header		
Source:	EuSubFunctionDataPort1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0..4	159:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
5	31:0	<b>Buffer Base Address</b>
		Format: <b>MHC_A32_BBA</b> Specifies the surface address offset page [31:10] for A32 stateless messages.
6	31:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
7	31:0	<b>Reserved</b>
		Access: RO
		Format: MBZ

## Stencil Message Data Payload Register

MDPR_STENCIL - Stencil Message Data Payload Register						
Size (in bits):	256					
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000					
DWord	Bit	Description				
0	31:24	<b>Stencil3</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>U8</td></tr></table> Stencil for Slot 3.		U8		
		U8				
	23:16	<b>Stencil2</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>U8</td></tr></table> Stencil for Slot 2.		U8		
		U8				
15:8	<b>Stencil1</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>U8</td></tr></table> Stencil for Slot 1.		U8			
	U8					
7:0	<b>Stencil0</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>U8</td></tr></table> Stencil for Slot 0.		U8			
	U8					
1	31:24	<b>Stencil7</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>U8</td></tr></table> Stencil for Slot 7.		U8		
		U8				
	23:16	<b>Stencil6</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>U8</td></tr></table> Stencil for Slot 6.		U8		
		U8				
15:8	<b>Stencil5</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>U8</td></tr></table> Stencil for Slot 5.		U8			
	U8					
7:0	<b>Stencil4</b> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>U8</td></tr></table> Stencil for Slot 4.		U8			
	U8					
2..7	191:0	<b>Reserved</b> Access: <table border="1" style="display: inline-table;"><tr><td> </td><td>RO</td></tr></table> Format: <table border="1" style="display: inline-table;"><tr><td> </td><td>MBZ</td></tr></table>		RO		MBZ
	RO					
	MBZ					



## Subset Atomic Integer Trinary Operation Message Descriptor Control Field

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

## Subset Reversed SIMD Mode 2 Message Descriptor Control Field

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



## Subset SIMD Mode 2 Message Descriptor Control Field

<b>MDC_SM2S - Subset SIMD Mode 2 Message Descriptor Control Field</b>											
Size (in bits):		1									
Default Value:		0x00000000									
DWord	Bit	Description									
0	0	<b>SIMD Mode</b> 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>SIMD8</td><td>SIMD8</td></tr><tr><td>01h</td><td>Reserved</td><td>Reserved.</td></tr></tbody></table>	Value	Name	Description	00h	SIMD8	SIMD8	01h	Reserved	Reserved.
Value	Name	Description									
00h	SIMD8	SIMD8									
01h	Reserved	Reserved.									



## Subset SIMD Mode 3 Message Descriptor Control Field

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



## Subspan Render Target Message Header Control

<b>MHC_RT_SUBSPAN - Subspan Render Target Message Header Control</b>		
Size (in bits):		32
Default Value:		0x00000000
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0	31:16	<b>Y</b> Format: U16 Y coordinate for upper-left pixel of this subspan
	15:0	<b>X</b> Format: U16 X coordinate for upper-left pixel of this subspan

## Surface Binding Table Index Message Descriptor Control Field

<b>MDC_BTS - Surface Binding Table Index Message Descriptor Control Field</b>																				
Size (in bits): 8																				
Default Value: 0x00000000																				
DWord	Bit	Description																		
0	7:0	<p><b>Binding Table Index</b> Specifies the Binding Table index for the message, which must be a Surface State Model.</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>00h-0EFh</td> <td>BTS</td> <td>Index of Binding Table State Surfaces</td> </tr> <tr> <td>0F0h-0FAh</td> <td>Reserved</td> <td>Reserved for future use</td> </tr> <tr> <td>0FCh</td> <td>SSO_BINDLESS</td> <td>Specifies a Surface State Offset into the Bindless Surface State heap, supplied by the extended message descriptor.</td> </tr> <tr> <td>0FBh</td> <td>Reserved</td> <td></td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>For Render Target Views, the Binding Table index need to be confined to the 00h to 0Fh range if Binding Table is not in the 256B alignment (18:8) mode. In the 256B alignment mode, the Binding Table Index need to be confined to the 00h to 3Fh range if slice common register 3. state cache perf fix disabled is set to 1, the entire range of BTI is supported.</p>	Value	Name	Description	00h-0EFh	BTS	Index of Binding Table State Surfaces	0F0h-0FAh	Reserved	Reserved for future use	0FCh	SSO_BINDLESS	Specifies a Surface State Offset into the Bindless Surface State heap, supplied by the extended message descriptor.	0FBh	Reserved		Others	Reserved	Ignored
Value	Name	Description																		
00h-0EFh	BTS	Index of Binding Table State Surfaces																		
0F0h-0FAh	Reserved	Reserved for future use																		
0FCh	SSO_BINDLESS	Specifies a Surface State Offset into the Bindless Surface State heap, supplied by the extended message descriptor.																		
0FBh	Reserved																			
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																										
Size (in bits):		8																								
Default Value:		0x00000000																								
DWord	Bit	Description																								
0	7:0	<p><b>Binding Table Index</b> Specifies the surface for the message, either Surface State Model or Stateless.</p> <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>0F0h-0FAh</td> <td>Reserved</td> <td>Reserved for future use</td> </tr> <tr> <td>0FCh</td> <td>SSO_BINDLESS</td> <td>Specifies a Surface State Offset into the Bindless Surface State heap, supplied by the extended message descriptor.</td> </tr> <tr> <td>0FBh</td> <td>Reserved</td> <td></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> <p style="text-align: center;"><b>Restriction</b></p> <p>When using A32_A64_NC, SW must ensure that 2 threads do not both access the same cache line (64B)</p>	Value	Name	Description	00h-0EFh	BTS	Index of Binding Table State Surfaces	0F0h-0FAh	Reserved	Reserved for future use	0FCh	SSO_BINDLESS	Specifies a Surface State Offset into the Bindless Surface State heap, supplied by the extended message descriptor.	0FBh	Reserved		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																								
0F0h-0FAh	Reserved	Reserved for future use																								
0FCh	SSO_BINDLESS	Specifies a Surface State Offset into the Bindless Surface State heap, supplied by the extended message descriptor.																								
0FBh	Reserved																									
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																								

## Surface Pixel Mask Message Header

<b>MH1_BTS_PSM - Surface Pixel Mask Message Header</b>		
Source:	EuSubFunctionDataPort1	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0..6	223:0	<b>Reserved</b>
		Access: RO
		Format: MBZ
7	31:0	<b>Reserved</b>
		Access: RO
		Format: MBZ

## SW Generated BINDING\_TABLE\_STATE

SW Generated BINDING_TABLE_STATE				
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	<b>Surface State Pointer</b>		
		<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 <b>Surface State Base Address</b></p>	Format:	SurfaceStateOffset[31:6]
	Format:	SurfaceStateOffset[31:6]		
	5:0	<b>Reserved</b>		
<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>		Access:	RO	Format:
Access:	RO			
Format:	MBZ			

# SWSB

SWSB																
Size (in bits):		8														
Default Value:		0x00000000														
DWord	Bit	Description														
0	7	<b>Mode</b> This field specifies the way SWSB information is specified. SingleInfo: (Type, Value) or DualInfo: (Value, Value).														
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SingleInfo <b>[Default]</b></td> <td>SingleInfo. Specifies the SWSB information as (Type, Value) pair. Where Value can be RegDist or SBID information.</td> </tr> <tr> <td>1</td> <td>DualInfo</td> <td>DualInfo. Specifies the SWSB information as (RegDistValue, SBIDValue) pair.</td> </tr> </tbody> </table>	Value	Name	Description	0	SingleInfo <b>[Default]</b>	SingleInfo. Specifies the SWSB information as (Type, Value) pair. Where Value can be RegDist or SBID information.	1	DualInfo	DualInfo. Specifies the SWSB information as (RegDistValue, SBIDValue) pair.					
		Value	Name	Description												
	0	SingleInfo <b>[Default]</b>	SingleInfo. Specifies the SWSB information as (Type, Value) pair. Where Value can be RegDist or SBID information.													
	1	DualInfo	DualInfo. Specifies the SWSB information as (RegDistValue, SBIDValue) pair.													
	<b>6:4 Type</b> Exists If: <span style="border: 1px solid black; padding: 2px;">([Mode]='SingleInfo')</span> This field specifies the type of SWSB information.															
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>RegDist <b>[Default]</b></td> </tr> <tr> <td>001b</td> <td>Reserved</td> </tr> <tr> <td>010b</td> <td>SBID.dst</td> </tr> <tr> <td>011b</td> <td>SBID.src</td> </tr> <tr> <td>100b</td> <td>SBID.set</td> </tr> <tr> <td>[101b-111b]</td> <td>Reserved</td> </tr> </tbody> </table>		Value	Name	000b	RegDist <b>[Default]</b>	001b	Reserved	010b	SBID.dst	011b	SBID.src	100b	SBID.set	[101b-111b]	Reserved
	Value	Name														
	000b	RegDist <b>[Default]</b>														
	001b	Reserved														
010b	SBID.dst															
011b	SBID.src															
100b	SBID.set															
[101b-111b]	Reserved															
<b>6:4 RegDistInfo</b> Exists If: <span style="border: 1px solid black; padding: 2px;">([Mode]='DualInfo')</span>																
<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Reserved</td> </tr> <tr> <td>[1-7]</td> <td>RegDistInfo</td> </tr> </tbody> </table>		Value	Name	0	Reserved	[1-7]	RegDistInfo									
Value	Name															
0	Reserved															
[1-7]	RegDistInfo															
<b>3 Reserved</b> Exists If: <span style="border: 1px solid black; padding: 2px;">([Mode]='SingleInfo') AND ([Type]='RegDist')</span> Format: MBZ																
<b>3:0 SBIDValue</b> Exists If: <span style="border: 1px solid black; padding: 2px;">([Mode]='SingleInfo') AND (([Type]='SBID.dst') OR ([Type]='SBID.src') OR ([Type]='SBID.set'))</span>																

<b>SWSB</b>							
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0-15]</td> <td>SBIDInfo</td> </tr> </tbody> </table>	Value	Name	[0-15]	SBIDInfo		
Value	Name						
[0-15]	SBIDInfo						
3:0	<b>SBIDValue</b> Exists If: ([Mode]='DualInfo')						
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0-15]</td> <td>SBIDInfo</td> </tr> </tbody> </table>	Value	Name	[0-15]	SBIDInfo		
Value	Name						
[0-15]	SBIDInfo						
2:0	<b>RegDistValue</b> Exists If: ([Mode]='SingleInfo') AND ([Type]='RegDist')						
	<table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No Dependency</td> </tr> <tr> <td>[1-7]</td> <td>RegDistInfo</td> </tr> </tbody> </table>	Value	Name	0	No Dependency	[1-7]	RegDistInfo
Value	Name						
0	No Dependency						
[1-7]	RegDistInfo						



## SZ OM S0A SIMD8 Render Target Data Payload

### MDP\_RTW\_ZMA8 - SZ OM S0A SIMD8 Render Target Data Payload

Size (in bits):	1792	
Default Value:	0x00000000, 0x00000000,	
DWord	Bit	Description
0.0-0.7	255:0	<b>Source 0 Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source 0 Alpha
1.0-1.7	255:0	<b>oMask</b> Format: <b>MDPR_OMASK</b> Slots [7:0] oMask. Upper half ignored.
2.0-2.7	255:0	<b>Red</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Red
3.0-3.7	255:0	<b>Green</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Green
4.0-4.7	255:0	<b>Blue</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
5.0-5.7	255:0	<b>Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha
6.0-6.7	255:0	<b>Source Depth</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source Depth

## SZ OM S0A SIMD16 Render Target Data Payload

MDP_RTW_ZMA16 - SZ OM S0A SIMD16 Render Target Data Payload		
Size (in bits):	3328	
Default Value:	0x00000000, 0x00000000,	
DWord	Bit	Description
0.0-1.7	511:0	<b>Source 0 Alpha</b> Format: <b>MDP_DW_SIMD16</b> Slots [15:0] Source 0 Alpha
2.0-2.7	255:0	<b>oMask</b> Format: <b>MDPR_OMASK</b> Slots [15:0] oMask
3.0-4.7	511:0	<b>Red</b> Format: <b>MDP_DW_SIMD16</b> Slots [15:0] Red
5.0-6.7	511:0	<b>Green</b> Format: <b>MDP_DW_SIMD16</b> Slots [15:0] Green

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



## MDP\_RTW\_ZM8DS - SZ OM SIMD8 Dual Source Render Target Data Payload

5.0-5.7	255:0	<b>Src1 Red</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Red
6.0-6.7	255:0	<b>Src1 Green</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Green
7.0-7.7	255:0	<b>Src1 Blue</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Blue
8.0-8.7	255:0	<b>Src1 Alpha</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Alpha
9.0-9.7	255:0	<b>Source Depth</b>
		Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> 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		
Size (in bits):	1536	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>oMask</b> Format: <span style="border: 1px solid black; padding: 2px;">MDPR_OMASK</span> Slots [7:0] oMask. Upper half ignored.
1.0-1.7	255:0	<b>Red</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Red
2.0-2.7	255:0	<b>Green</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Green
3.0-3.7	255:0	<b>Blue</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Blue
4.0-4.7	255:0	<b>Alpha</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Alpha
5.0-5.7	255:0	<b>Source Depth</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] Source Depth





## MDP\_RTW\_ZM16 - SZ OM SIMD16 Render Target Data Payload

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



## SZ S0A SIMD8 Render Target Data Payload

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



## MDP\_RTW\_ZA16 - SZ S0A SIMD16 Render Target Data Payload

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



MDP_RTW_Z8DS - SZ SIMD8 Dual Source Render Target Data Payload		
6.0-6.7	255:0	<b>Src1 Blue</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Blue
7.0-7.7	255:0	<b>Src1 Alpha</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots[7:0] or [15:8] of Src1 Alpha
8.0-8.7	255:0	<b>Source Depth</b> Format: <span style="border: 1px solid black; padding: 2px;">MDP_DW_SIMD8</span> Slots [7:0] or [15:8] of Source Depth



## SZ SIMD8 Render Target Data Payload

MDP_RTW_Z8 - SZ SIMD8 Render Target Data Payload		
Size (in bits):	1280	
Default Value:	0x00000000, 0x00000000	
DWord	Bit	Description
0.0-0.7	255:0	<b>Red</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Red
1.0-1.7	255:0	<b>Green</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Green
2.0-2.7	255:0	<b>Blue</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
3.0-3.7	255:0	<b>Alpha</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha
4.0-4.7	255:0	<b>Source Depth</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source Depth

## SZ SIMD16 Render Target Data Payload

<b>MDP_RTW_Z16 - SZ SIMD16 Render Target Data Payload</b>		
Size (in bits):	2560	
Default Value:	0x00000000, 0x00000000,	
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0.0-0.7	255:0	<b>Red[7:0]</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Red
1.0-1.7	255:0	<b>Red[15:8]</b> Format: <b>MDP_DW_SIMD8</b> Slots [15:8] Red
2.0-2.7	255:0	<b>Green[7:0]</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Green
3.0-3.7	255:0	<b>Green[15:8]</b> Format: <b>MDP_DW_SIMD8</b> Slots [15:8] Green
4.0-4.7	255:0	<b>Blue[7:0]</b> Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Blue
5.0-5.7	255:0	<b>Blue[15:8]</b> Format: <b>MDP_DW_SIMD8</b> Slots [15:8] Blue

MDP_RTW_Z16 - SZ SIMD16 Render Target Data Payload		
6.0-6.7	255:0	<b>Alpha[7:0]</b>
		Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Alpha
7.0-7.7	255:0	<b>Alpha[15:8]</b>
		Format: <b>MDP_DW_SIMD8</b> Slots [15:8] Alpha
8.0-8.7	255:0	<b>Source Depth[7:0]</b>
		Format: <b>MDP_DW_SIMD8</b> Slots [7:0] Source Depth
9.0-9.7	255:0	<b>Source Depth[15:8]</b>
		Format: <b>MDP_DW_SIMD8</b> Slots [15:8] Source Depth



## 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	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:25	<b>Message Length</b>
		Default Value: 1h One GRF
		Format: U4
	24:20	<b>Response Length</b>
		Default Value: 0h Zero GRF
		Format: U5
	19	<b>Header Present</b>
		Format: MBZ
	18:1	<b>Reserved</b>
		Access: RO
		Format: MBZ
0	<b>Message Type</b>	
	Default Value: 0h End Thread	
	Format: Opcode	
	End of Thread message opcode	



## TILE\_RECT

TILE_RECT				
Source:	RenderCS, PositionCS			
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:16	<b>Tile Rectangle Y Min</b>		
		Format: U16		
		Specifies Y Min coordinate of (inclusive) Tile Region used for tile rendering test.		
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Value	Name
Value	Name			
[0,16383]				
15:0	15:0	<b>Tile Rectangle X Min</b>		
		Format: U16		
		Specifies X Min coordinate of (inclusive) Tile Region used for tile rendering test.		
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Value	Name
Value	Name			
[0,16383]				
1	31:16	<b>Tile Rectangle Y Max</b>		
		Format: U16		
		Specifies Y Max coordinate of (inclusive) Tile Region used for tile rendering test.		
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Value	Name
Value	Name			
[0,16383]				
15:0	15:0	<b>Tile Rectangle X Max</b>		
		Format: U16		
		Specifies X Max coordinate of (inclusive) Tile Region used for tile rendering test.		
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> </tr> </tbody> </table>	Value	Name
Value	Name			
[0,16383]				

## Timeout Data Payload

<b>MDP_TIMEOUT - Timeout Data Payload</b>		
Source:	EuSubFunctionGateway	
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:10	<b>Reserved</b>
		Access: RO
		Format: MBZ
	9:0	<b>Timeout Value</b>
	Format: U10	
	<b>Description</b>	
	The amount of time GW should wait before sending a writeback message. This value is in terms of 1024 clocks. Thus, with a 1Ghz clock it would be approximately in terms of uS. 0 and 1 are illegal values since the actual timeout time can be short by up to 1 increment of the timeout value.	
1..7	223:0	<b>Reserved</b>
		Access: RO
		Format: MBZ



## U24\_X8

U24_X8 - U24_X8		
Size (in bits):	32	
Default Value:	0x00000000	
32bit packed 24Unorm and 8bit of unused format (named X8)		
DWord	Bit	Description
0	31:8	<b>MSB_U24</b> Format: U24
	7:0	<b>LSB_X8</b> Format: U8

## Untyped Write Channel Mask Message Descriptor Control Field

<b>MDC_UW_CMASK - Untyped Write Channel Mask Message Descriptor Control Field</b>																				
Size (in bits):		4																		
Default Value:		0x00000000																		
DWord	Bit	Description																		
0	3:0	<p><b>Mask</b> For untyped surface write messages, indicates which channels are included in the message payload and written to the surface.</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>00h</td> <td>RGBA <b>[Default]</b></td> <td>Red, Green, Blue, and Alpha are included</td> </tr> <tr> <td>08h</td> <td>RGB</td> <td>Red, Green, and Blue are included</td> </tr> <tr> <td>0Ch</td> <td>RG</td> <td>Red and Green are included</td> </tr> <tr> <td>0Eh</td> <td>R</td> <td>Red is included</td> </tr> <tr> <td>Others</td> <td>Reserved</td> <td>Ignored</td> </tr> </tbody> </table>	Value	Name	Description	00h	RGBA <b>[Default]</b>	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
Value	Name	Description																		
00h	RGBA <b>[Default]</b>	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			
Size (in bits):	256		
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000		
DWord	Bit	Description	
0.0-0.3	127:0	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
0.4-0.7	127:0	<b>Oword</b>	
		Format:	U128
Specifies the upper Oword data element			

## URB Channel Mask Payload Control

MACD_URB_CMASK - URB Channel Mask Payload Control		
Size (in bits):		32
Default Value:		0x00000000
DWord	Bit	Description
0	31:24	<b>Reserved</b>
		Access: RO
		Format: MBZ
	23:16	<b>Channel Mask</b>
		Format: Enable[8] For each channel present in the message data payload, the corresponding channel mask bit is ANDed with the slot's execution mask to determine the final channel enable. When final channel enable is 1 it indicates that Dword data will be written to the surface.
	15:0	<b>Reserved</b>
		Access: RO
		Format: MBZ



## URB Handle Message Header

<b>MH_URB_HANDLE - URB Handle Message Header</b>		
Size (in bits):	256	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0.0 This is the URB handle where slot 0 results are written or read.	31:0	<b>Handle 0</b> Format: <b>MHC_URB_HANDLE</b>
0.1 This is the URB handle where slot 1 results are written or read.	31:0	<b>Handle 1</b> Format: <b>MHC_URB_HANDLE</b>
0.2 This is the URB handle where slot 2 results are written or read.	31:0	<b>Handle 2</b> Format: <b>MHC_URB_HANDLE</b>
0.3 This is the URB handle where slot 3 results are written or read.	31:0	<b>Handle 3</b> Format: <b>MHC_URB_HANDLE</b>
0.4 This is the URB handle where slot 4 results are written or read.	31:0	<b>Handle 4</b> Format: <b>MHC_URB_HANDLE</b>
0.5 This is the URB handle where slot 5 results are written or read.	31:0	<b>Handle 5</b> Format: <b>MHC_URB_HANDLE</b>
0.6 This is the URB handle where slot 6 results are written or read.	31:0	<b>Handle 6</b> Format: <b>MHC_URB_HANDLE</b>
0.7 This is the URB handle where slot 7 results are written or read.	31:0	<b>Handle 7</b> Format: <b>MHC_URB_HANDLE</b>



## URB Handle Message Header Control

<b>MHC_URB_HANDLE - URB Handle Message Header Control</b>		
Size (in bits):	32	
Default Value:	0x00000000	
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0	31:25	<b>Reserved</b>
		Access: RO
	Format: MBZ	
	24:0	<b>Reserved</b>
Access: RO		
Format: MBZ		

## VC1

<b>VC1</b>						
Source:	VideoCS					
Size (in bits):	16					
Default Value:	0x00000000					
DWord	Bit	Description				
0	15:8	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="text-align: center;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	7	<b>Syncmarker Error</b> This flag indicates missing sync marker SEs coded in the bit-stream.				
	6	<b>Mbmode SE Error</b> This flag indicates inconsistent Macroblock SEs coded in the bit-stream.				
	5	<b>Transformtype SE Error</b> This flag indicates inconsistent transform type SEs coded in the bit-stream.				
	4	<b>Coefficient Error</b> This flag indicates inconsistent Coefficient SEs coded in the bit-stream.				
	3	<b>Motion Vector SE Error</b> This flag indicates inconsistent Motion Vector SEs coded in the bit-stream.				
	2	<b>Coded Block Pattern CY SE Error</b> This flag indicates inconsistent CBPCY SEs coded in the bit-stream.				
	1	<b>Mquant Error</b> This flag indicates inconsistent MQANT SEs coded in the bit-stream.				
0	<b>MB Concealment Flag</b> . 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	15:12	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	11	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	10:3	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	2	<b>Command Privilege Violation Error</b> 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	<b>Reserved</b>	
		Access:	RO
Format:		MBZ	
0	<b>Instruction Error</b> 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"> <li>Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported).</li> <li>Defeatured MI Instruction Opcodes:</li> </ul>		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	1		Instruction Error detected
	<b>Programming Notes</b>		
	This error indications cannot be cleared except by reset (i.e., it is a fatal error).		

## VD\_CONTROL\_STATE\_BODY

VD_CONTROL_STATE_BODY			
Size (in bits):	64		
Default Value:	0x00000000, 0x00000000		
DWord	Bit	Description	
0	31:30	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	29	<b>Reserved</b>	
	28	<b>Reserved</b>	
	27:1	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	0	<b>Pipeline Initialization</b>	
		<b>Description</b>	
This bit, when set, clears internal states for HCP Pipe if Media Instruction Opcode is set for HCP Pipe..			
This bit, when set, clears internal states for AVP Pipe if Media Instruction Opcode is set for AVP Pipe.			
1	31:3	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	2	<b>Memory Implicit Flush</b> This is used to initiate an implicit flush to memory to make sure all the memory request goes to memory. This should be programmed at the end of each frame after frame completion and before MI_FLUSH.	
	1	<b>Scalable Mode Pipe Unlock</b> This is used for decoder/encoder pipe to unlock all the pipes for scalable mode. It should be programmed at the end of frame.	
	0	<b>Scalable Mode Pipe Lock</b> This is used for decoder/encoder pipe to lock all the pipes for scalable mode. It should be programmed at the start of frame.	

## VDENC\_64B\_Aligned\_Lower\_Address

VDENC_64B_Aligned_Lower_Address		
Source:	VideoCS	
Size (in bits):	32	
Default Value:	0x00000000	
DWord	Bit	Description
0	31:6	<b>Address</b>
		Format: GraphicsAddress[31:6]
		[31:6] This field is for the 26 bits of the lower address.
	5:0	<b>Reserved</b>
		Access: RO Format: MBZ



## VDENC\_64B\_Aligned\_Upper\_Address

VDENC_64B_Aligned_Upper_Address			
Source:	VideoCS		
Size (in bits):	32		
Default Value:	0x00000000		
DWord	Bit	Description	
0	31:16	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	15:0	<b>Address Upper DWord</b> Bits [47:32] of the Address. GraphicsAddress is 64-bit value [63:0], but only a portion of it is used by hardware. The uppermost bits [63:48] are ignored and MBZ.	

## VDENC\_Block\_8x8\_4

VDENC_Block_8x8_4		
Source:	VideoCS	
Size (in bits):	16	
Default Value:	0x00000000	
DWord	Bit	Description
0	15:12	<b>Block8x8[3]</b> Format: U4
	11:8	<b>Block8x8[2]</b> Format: U4
	7:4	<b>Block8x8[1]</b> Format: U4
	3:0	<b>Block8x8[0]</b> Format: U4



## VDENC\_Colocated\_MV\_Picture

VDENC_Colocated_MV_Picture		
Source:	VideoCS	
Size (in bits):	96	
Default Value:	0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	<b>Lower Address</b>
		Format: <b>VDENC_64B_Aligned_Lower_Address</b>
		64 byte aligned buffer.
		This field is used to write the DMV data by VDEnc. VDEnc only supports spatial direct prediction and not temporal direct. Hence the HW precomputes the ColZeroFlag per 8x8 block and writes 8-bits per macroblock. HW accumulates a CL worth of data before writing it out. This is a linear buffer, can be considered to be a frame level row-store. There is no read/write happening to the surface for any given frame.
		HW only writes to this surface for P-Frames.
		HW only reads from this surface for B-Frames.
		Size = 8-bits/ MB linear buffer.
1	31:0	<b>Upper Address</b>
		Format: <b>VDENC_64B_Aligned_Upper_Address</b>
2	31:0	<b>Picture Fields</b>
		Format: <b>VDENC_Surface_Control_Bits</b>



## VDENC\_Delta\_MV\_XY

VDENC_Delta_MV_XY							
Source:		VideoCS					
Size (in bits):		128					
Default Value:		0x00000000, 0x00000000, 0x00000000, 0x00000000					
<b>16x16</b>	<b>16x08</b>	<b>08x16</b>		<b>08x08</b>			
0	1	3	4	5	6		
	2			7	8		
Calculates the difference between the actual MV for the Sub Macroblock and the predicted MV based on the availability of the neighbors.							
Programming Notes							
This is calculated and populated for Inter frames only. In case of an Intra MB in Inter frames, this value should be 0.							
DWord	Bit	Description					
0..3	127:112	<b>Y3</b>					
		Default Value:					0
		Format:					U16
		<b>Mb_type</b>	<b>PartID</b>				
	8x8	8					
	111:96	<b>X3</b>					
		Default Value:					0
		Format:					U16
		<b>Mb_type</b>	<b>PartID</b>				
	8x8	8					
	95:80	<b>Y2</b>					
		Default Value:					0
		Format:					U16
		<b>Mb_type</b>	<b>PartID</b>				
	8x8	7					
	79:64	<b>X2</b>					
		Default Value:					0
		Format:					U16

## VDENC\_Delta\_MV\_XY

VDENC_Delta_MV_XY															
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Mb_type</th> <th style="text-align: center;">PartID</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">8x8</td> <td style="text-align: center;">7</td> </tr> </tbody> </table>	Mb_type	PartID	8x8	7										
Mb_type	PartID														
8x8	7														
63:48	<p><b>Y1</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U16</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Mb_type</th> <th style="text-align: center;">PartID</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">16x8</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">8x16</td> <td style="text-align: center;">4</td> </tr> <tr> <td style="text-align: center;">8x8</td> <td style="text-align: center;">6</td> </tr> </tbody> </table>	Default Value:	0	Format:	U16	Mb_type	PartID	16x8	2	8x16	4	8x8	6		
Default Value:	0														
Format:	U16														
Mb_type	PartID														
16x8	2														
8x16	4														
8x8	6														
47:32	<p><b>X1</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U16</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Mb_type</th> <th style="text-align: center;">PartID</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">16x8</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">8x16</td> <td style="text-align: center;">4</td> </tr> <tr> <td style="text-align: center;">8x8</td> <td style="text-align: center;">6</td> </tr> </tbody> </table>	Default Value:	0	Format:	U16	Mb_type	PartID	16x8	2	8x16	4	8x8	6		
Default Value:	0														
Format:	U16														
Mb_type	PartID														
16x8	2														
8x16	4														
8x8	6														
31:16	<p><b>Y0</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U16</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Mb_type</th> <th style="text-align: center;">PartID</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">16x16</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">16x8</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">8x16</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">8x8</td> <td style="text-align: center;">5</td> </tr> </tbody> </table>	Default Value:	0	Format:	U16	Mb_type	PartID	16x16	0	16x8	1	8x16	3	8x8	5
Default Value:	0														
Format:	U16														
Mb_type	PartID														
16x16	0														
16x8	1														
8x16	3														
8x8	5														
15:0	<p><b>X0</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Default Value:</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U16</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Mb_type</th> <th style="text-align: center;">PartID</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">16x16</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">16x8</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">8x16</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">8x8</td> <td style="text-align: center;">5</td> </tr> </tbody> </table>	Default Value:	0	Format:	U16	Mb_type	PartID	16x16	0	16x8	1	8x16	3	8x8	5
Default Value:	0														
Format:	U16														
Mb_type	PartID														
16x16	0														
16x8	1														
8x16	3														
8x8	5														

## VDENC\_Down\_Scaled\_Reference\_Picture

VDENC_Down_Scaled_Reference_Picture				
Source:	VideoCS			
Size (in bits):	96			
Default Value:	0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	<b>Lower Address</b> <table border="1" data-bbox="321 611 1466 657"> <tr> <td>Format:</td> <td><b>VDENC_64B_Aligned_Lower_Address</b></td> </tr> </table> <p>Specifies the 64 byte aligned DownScaled reference frame buffer address. VDEnc supports upto 3 down-scaled reference pictures for HME search. (2 fwd and 1 bwd).</p>	Format:	<b>VDENC_64B_Aligned_Lower_Address</b>
Format:	<b>VDENC_64B_Aligned_Lower_Address</b>			
1	31:0	<b>Upper Address</b> <table border="1" data-bbox="321 810 1466 856"> <tr> <td>Format:</td> <td><b>VDENC_64B_Aligned_Upper_Address</b></td> </tr> </table>	Format:	<b>VDENC_64B_Aligned_Upper_Address</b>
Format:	<b>VDENC_64B_Aligned_Upper_Address</b>			
2	31:0	<b>Picture Fields</b> <table border="1" data-bbox="321 905 1466 951"> <tr> <td>Format:</td> <td><b>VDENC_Surface_Control_Bits</b></td> </tr> </table>	Format:	<b>VDENC_Surface_Control_Bits</b>
Format:	<b>VDENC_Surface_Control_Bits</b>			



## VDENC\_Original\_Uncompressed\_Picture

VDENC_Original_Uncompressed_Picture		
Source:	VideoCS	
Size (in bits):	96	
Default Value:	0x00000000, 0x00000000, 0x00000000	
DWord	Bit	Description
0	31:0	<b>Lower Address</b>
		Format: <b>VDENC_64B_Aligned_Lower_Address</b>
		Specifies the 64 byte aligned frame buffer address for fetching YUV pixel data from the originaluncompressed input picture for encoding.
		This field is only valid in encoding mode.
1	31:0	<b>Upper Address</b>
		Format: <b>VDENC_64B_Aligned_Upper_Address</b>
2	31:0	<b>Picture Fields</b>
		Format: <b>VDENC_Surface_Control_Bits</b>

## VDENC\_Reference\_Picture

VDENC_Reference_Picture				
Source:	VideoCS			
Size (in bits):	96			
Default Value:	0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0	31:0	<p><b>Lower Address</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>VDENC_64B_Aligned_Lower_Address</b></td> </tr> </table> <p>Specifies the 64 byte aligned reference frame buffer addresses corresponding to fwd reference index = 0 in the bitstream. VDEnc supports upto 4 reference pictures for IME search. (3 fwd and 1 bwd).</p>	Format:	<b>VDENC_64B_Aligned_Lower_Address</b>
Format:	<b>VDENC_64B_Aligned_Lower_Address</b>			
1	31:0	<p><b>Upper Address</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>VDENC_64B_Aligned_Upper_Address</b></td> </tr> </table>	Format:	<b>VDENC_64B_Aligned_Upper_Address</b>
Format:	<b>VDENC_64B_Aligned_Upper_Address</b>			
2	31:0	<p><b>Picture Fields</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>VDENC_Surface_Control_Bits</b></td> </tr> </table>	Format:	<b>VDENC_Surface_Control_Bits</b>
Format:	<b>VDENC_Surface_Control_Bits</b>			



## VDENC\_Row\_Store\_Scratch\_Buffer\_Picture

VDENC_Row_Store_Scratch_Buffer_Picture		
Source:	VideoCS	
Size (in bits):	96	
Default Value:	0x00000000, 0x00000000, 0x00000000	
Structure_VDENC_Row_Store_Scratch_Buffer_Picture: true		
DWord	Bit	Description
0	31:0	<b>Lower Address</b>
		Format: <b>VDENC_64B_Aligned_Lower_Address</b>
		<p>This field provides the base address of the scratch buffer (read/write) used by VDEnc to store MB information of the previous row for processing of each macroblock in the current row. The Row Store buffer must be 64-byte cacheline aligned. Hardware uses the horizontal address of the current macroblock to address the Row Store.</p> <p><b>AVC:</b> Size = 256 bits/MB. 4K wide picture needs 128 CLs.</p>
1	31:0	<b>Upper Address</b>
		Format: <b>VDENC_64B_Aligned_Upper_Address</b>
2	31:0	<b>Buffer Picture Fields</b>
		Format: <b>VDENC_Surface_Control_Bits</b>

## VDENC\_Sub\_Mb\_Pred\_Mode

VDENC_Sub_Mb_Pred_Mode		
Source:	VideoCS	
Size (in bits):	8	
Default Value:	0x00000000	
DWord	Bit	Description
0	7:6	<b>SubMbPredMode[3]</b> Format: U2
	5:4	<b>SubMbPredMode[2]</b> Format: U2
	3:2	<b>SubMbPredMode[1]</b> Format: U2
	1:0	<b>SubMbPredMode[0]</b> Format: U2

## VDENC\_Surface\_Control\_Bits

VDENC_Surface_Control_Bits																	
Source:	VideoCS																
Size (in bits):	32																
Default Value:	0x00000000																
DWord	Bit	Description															
0	31:21	<b>Reserved</b> Format: MBZ															
	20:16	<b>Reserved</b> Format: MBZ															
	15	<b>Reserved</b> Format: MBZ															
	14:13	<b>Tiled Resource Mode</b> Format: U2  <b>For Media Surfaces:</b> This field specifies the tiled resource mode. Only supports 4K tiling for all surfaces.  <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															
			<b>Programming Notes</b>														
		This field should be programmed the same for all these VDEnc surfaces listed below. <b>DS FWD REF0, DS FWD REF1, DS BWD REF0, FWD REF0, FWD REF1, FWD REF2, BWD REF0.</b>															
12		<b>Cache Select</b> Exists If: (Property[Structure_VDENC_Row_Store_Scratch_Buffer_Picture] == 'true') Format: U1															
		<b>Description</b>															
		This field controls if the Row Store is going to store inside Media Cache (rowstore cache) or to LLC.															
		When this is programmed to "1" (going to Media Cache), the corresponding base address will be programmed with the starting position in the media cache. The programming table is in "Buffer Size Requirement Page" in HEVC section															



VDENC_Surface_Control_Bits		
Value	Name	Description
0		Buffer going to LLC.
1		Buffer going to Internal Media Storage.
11	<b>Reserved</b>	
Access:		RO
Format:		MBZ
10	<b>Compression Type</b>	
Value	Name	
0	Media Compression Enabled <b>[Default]</b>	
1	Render Compression Enabled	
9	<b>Memory Compression Enable</b>	
Format:		Enable
Memory compression will be attempted for this surface.		
Value	Name	
0h	Disable	
1h	Enable	
<b>Programming Notes</b>		
This field should be programmed the same for all these VDEnc surfaces listed below. <b>DS FWD REF0, DS FWD REF1, DS BWD REF0, FWD REF0, FWD REF1, FWD REF2, BWD REF0.</b>		
8:7	<b>Arbitration Priority Control</b>	
Format:		U2
This field controls the priority of arbitration used in the GAC/GAM pipeline for this surface.		
Value	Name	Description
00b	Highest priority	
01b	Second highest priority	
10b	Third highest priority	
11b	Lowest priority	
6:1	<b>Index to Memory Object Control State (MOCS) Tables:</b>	
Format:		U6
The index to define the L3 and system cache memory properties. The details of the controls are further defined in L3 and Page walker (memory interface) control registers.		
The field is defined to populate 64 different surface controls to be used concurrently. Related control registers can be updated during runtime.		
0	<b>Reserved</b>	



## VDENC\_Surface\_State\_Fields

VDENC_Surface_State_Fields								
Source:	VideoCS							
Size (in bits):	128							
Default Value:	0x00000000, 0x00000003, 0x00000000, 0x00000000							
DWord	Bit	Description						
0	31:18	<b>Height</b> Format: U14-1 This field specifies the height of the Picture 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> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> <td>Representing heights [1,16384]</td> </tr> </tbody> </table>	Value	Name	Description	[0,16383]		Representing heights [1,16384]
		Value	Name	Description				
[0,16383]		Representing heights [1,16384]						
<b>Programming Notes</b> AVC specific Note: <ul style="list-style-type: none"> <li>When surface tiling is TileY, the Frame Height needs to be programmed as 16-pixel aligned.</li> <li>When surface tiling is Linear (supported only for Source surface), the Frame Height can be programmed 2-pixel aligned.</li> </ul>								
	17:4	<b>Width</b> Format: U14-1 This field specifies the width of the Picture in units of pixels/residuals. For PLANAR surface formats, this field indicates the width of the Y (luma) plane.						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0,16383]</td> <td></td> <td>Representing widths [1,16384]</td> </tr> </tbody> </table>	Value	Name	Description	[0,16383]		Representing widths [1,16384]
		Value	Name	Description				
[0,16383]		Representing widths [1,16384]						
<b>Programming Notes</b> <ul style="list-style-type: none"> <li>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).</li> <li>Width (field value + 1) must be a multiple of 2 for PLANAR_420,</li> <li>VDenc HW does not use this field, the picture width is read from IMG State instead, because this field may not equal to the actual picture width. This field is used by the KMD to allocate surface in GTT.</li> </ul>								
	3	<b>Color space selection</b> Format: U1						
		<table border="1"> <tr> <td>0</td> <td>Use BT.601 Space conversion</td> </tr> <tr> <td>1</td> <td>Use BT.709 Space conversion</td> </tr> </table>	0	Use BT.601 Space conversion	1	Use BT.709 Space conversion		
		0	Use BT.601 Space conversion					
1	Use BT.709 Space conversion							

<b>VDENC_Surface_State_Fields</b>																																				
2	<b>Reserved</b>																																			
	Access: RO																																			
	Format: MBZ																																			
1:0	<b>Cr(V)/Cb(U) Pixel Offset V Direction</b>																																			
	Format: U0.2																																			
	Exactly as shown in the original spec.																																			
	Specifies the distance to the U/V values with respect to the even numbered Y channels in the V direction.																																			
	<b>Programming Notes</b>																																			
This field is ignored for all formats except PLANAR_420_8.																																				
1	31:27 <b>Reserved</b>																																			
	Access: RO																																			
	Format: MBZ																																			
	26:23 <b>Reserved</b>																																			
	Access: RO																																			
	Format: MBZ																																			
	22:20 <b>Chroma Downsample Filter Control</b>																																			
	Format: U3																																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Value</th> <th>Left Side Tap</th> <th>Center Tap</th> <th>Right Side Tap</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>64</td><td>0</td></tr> <tr><td>1</td><td>16</td><td>32</td><td>16</td></tr> <tr><td>2</td><td>15</td><td>34</td><td>15</td></tr> <tr><td>3</td><td>14</td><td>36</td><td>14</td></tr> <tr><td>4</td><td>13</td><td>38</td><td>13</td></tr> <tr><td>5</td><td>12</td><td>40</td><td>12</td></tr> <tr><td>6</td><td>11</td><td>42</td><td>11</td></tr> <tr><td>7</td><td>0</td><td>32</td><td>32</td></tr> </tbody> </table>	Value	Left Side Tap	Center Tap	Right Side Tap	0	0	64	0	1	16	32	16	2	15	34	15	3	14	36	14	4	13	38	13	5	12	40	12	6	11	42	11	7	0	32	32
Value	Left Side Tap	Center Tap	Right Side Tap																																	
0	0	64	0																																	
1	16	32	16																																	
2	15	34	15																																	
3	14	36	14																																	
4	13	38	13																																	
5	12	40	12																																	
6	11	42	11																																	
7	0	32	32																																	
<b>Programming Notes</b>																																				
For Tile Y 444 -> 420, Filter settings on 0 and 7 are valid. All other combinations are invalid. This is true for 10 bit and 8 bit.																																				
19:3 <b>Surface Pitch</b>																																				
Format: U17																																				
-1 pitch in Bytes																																				

## VDENC\_Surface\_State\_Fields

		<p>This field specifies the surface pitch in (#Bytes).</p>	
		<b>Programming Notes</b>	
		<p>For tiled surfaces, the pitch must be a multiple of the tile width (i.e.128 bytes aligned). If Half Pitch for Chroma is set, this field must be a multiple of two tile widths for tiled surfaces, or a multiple of 2 bytes for linear surfaces. For Y-tiled surfaces: Range = [127,131071] to [128B,128KB] = [1 tile, 1028 tiles].</p>	
		<p>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 <math>[2^{Cu}-1, 131071] \rightarrow [(2^{Cu})B, 128KB] = [1 \text{ tile}, 128KB/(2^{Cu} \text{ tiles})]</math></p> <p>The field specifies the surface pitch in (#Bytes - 1)</p>	
2	<b>Half Pitch for Chroma</b>	Format:	Enable
		<p>(This field must be set to Disable.) This field indicates that the chroma plane(s) will use a pitch equal to half the value specified in the Surface Pitch field. This field is only used for PLANAR surface formats. This field is ignored by VDEnc (unless we support YV12).</p>	
		<b>Value</b>	<b>Name</b>
		0	Disable <b>[Default]</b>
		1	Enable
1	<b>Tiled Surface</b>	Format:	Boolean
		<p>(This field must be set to TRUE: Tiled.) This field specifies whether the surface is tiled. This field is ignored by VDEnc usage.</p>	
		<b>Value</b>	<b>Name</b>
		0	False
		1	True <b>[Default]</b>
		<b>Programming Notes</b>	
		<p>Linear surfaces can be mapped to Main Memory (uncached) or System Memory (cacheable, snooped).Tiled surfaces can only be mapped to Main Memory. The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this bit.</p>	
0	<b>Tile Walk</b>	Format:	Boolean
		<p>(This field must be set to 1: TILEWALK_YMAJOR.) This field specifies the type of memory tiling(XMajor or YMajor) employed to tile this surface. See Memory Interface Functions for detailson memory tiling and restrictions.This field is ignored when the surface is linear. InternallyH/W always treats this as set to 1 for all VDEnc usage.</p>	
		<b>Value</b>	<b>Name</b>
		0h	XMAJOR
			TILEWALK_XMAJOR

VDENC_Surface_State_Fields		
	1h	YMAJOR [Default] TILEWALK_YMAJOR
	<b>Programming Notes</b>	
	The corresponding cache(s) must be invalidated before a previously accessed surface is accessed again with an altered state of this bit.	
2	31	<b>Reserved</b>
		Access: RO
		Format: MBZ
	30:16	<b>X Offset for U(Cb)</b>
		Format: U15
		Pixel Offset This field specifies the horizontal offset in pixels from the Surface Base Address to the start(origin) of the U(Cb) plane or the interleaved UV plane if Interleave Chroma is enabled. This field is only used for PLANAR surface formats. This field must be set to zero. X Offset for U(Cb) in pixel. (This field must be zero for NV12 and IMC 1 and 3).
	<b>Programming Notes</b>	
	For PLANAR_420 and PLANAR_422 surface formats, this field must be zero.	
15		<b>Reserved</b>
		Access: RO
		Format: MBZ
	14:0	<b>Y Offset for U(Cb)</b>
		Format: U15
		Pixel Row Offset This field specifies the vertical offset in rows from the Surface Base Address to the start(origin) of the U(Cb) plane or the interleaved UV plane if Interleave Chroma is enabled. This field is only used for PLANAR surface formats.
	<b>Programming Notes</b>	
	For PLANAR_420 and PLANAR_422 surface formats, this field must be multiple of 8 pixels - i.e. multiple MBs.	
3	31:29	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:16	<b>X Offset for V(Cr)</b>
	Format: U13	

<b>VDENC_Surface_State_Fields</b>			
	<p>Offset in Pixels</p> <p>This field must be zero for NV12 and IMC 1 and 3.</p> <p>This field specifies the horizontal offset in pixels from the Surface Base Address to the start (origin) of the V(Cr) plane. This field is only used for PLANAR surface formats with Interleave Chroma disabled.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>For PLANAR_420 and PLANAR_422 surface formats, this field must indicate an even number of pixels.</p>		
15:0	<p><b>Y Offset for V(Cr)</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td style="width: 40%;">U16</td> </tr> </table> <p>Row Offset in Pixels</p> <p>This field specifies the vertical offset in rows from the Surface Base Address to the start (origin) of the V(Cr) plane. This field is only used for PLANAR surface formats with Interleave Chroma disabled. This field is ignored by all video codec, only used by JPEG.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p>For PLANAR_420 surface formats, this field must be multiple of 16 pixels - i.e. multiple MBs.</p>	Format:	U16
Format:	U16		

## VEBOX\_ACE\_LACE\_STATE

VEBOX_ACE_LACE_STATE																	
Source:	VideoEnhancementCS																
Size (in bits):	992																
Default Value:	0x00000068, 0x4C382410, 0x9C887460, 0xEBD8C4B0, 0x604C3824, 0xB09C8874, 0x0000D8C4, 0x04000400, 0x04000400, 0x04000400, 0x04000400, 0x04000400, 0x00000400, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000																
This state structure contains the IECF State Table Contents for ACE state.																	
DWord	Bit	Description															
0	31:16	<b>Min_ACE_luma</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U16</td> </tr> </table>	Format:	U16													
	Format:	U16															
	15:14	<b>LACE Single Histogram Set</b> This bit tells LACE which frames will be included in the histogram when the Deinterlacer is enabled. <table border="1" style="width: 100%;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Current</td> <td>The histogram includes only the current frame.</td> </tr> <tr> <td>01b</td> <td>Previous</td> <td>The histogram includes only the previous frame.</td> </tr> <tr> <td>10b</td> <td>Current + Previous</td> <td>The histogram includes pixels from both the current and previous frame.</td> </tr> <tr> <td>11b</td> <td>Previous + Current</td> <td>The histogram includes the previous frame followed by the current frame.</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> When the Deinterlacer is disabled, this field must be 00b.If <b>DI Output Frames</b> 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	Value	Name	Description	00b	Current	The histogram includes only the current frame.	01b	Previous	The histogram includes only the previous frame.	10b	Current + Previous	The histogram includes pixels from both the current and previous frame.	11b	Previous + Current	The histogram includes the previous frame followed by the current frame.
	Value	Name	Description														
	00b	Current	The histogram includes only the current frame.														
	01b	Previous	The histogram includes only the previous frame.														
10b	Current + Previous	The histogram includes pixels from both the current and previous frame.															
11b	Previous + Current	The histogram includes the previous frame followed by the current frame.															
13	<b>LACE Histogram Size</b> <table border="1" style="width: 100%;"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>128-bin histogram</td> </tr> <tr> <td>1</td> <td>256-bin histogram</td> </tr> </tbody> </table>	Value	Name	0	128-bin histogram	1	256-bin histogram										
Value	Name																
0	128-bin histogram																
1	256-bin histogram																
12	<b>LACE Histogram Enable</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>0</td> </tr> </table> This bit enables the collection of LACE histogram data. If this bit is 0 then only the ACE histogram will be collected.	Default Value:	0														
Default Value:	0																

VEBOX_ACE_LACE_STATE									
	11:7	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
	Access:	RO							
	Format:	MBZ							
	6:2	<b>Skin Threshold</b> <table border="1"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <p>Used for Y analysis (min/max) for pixels which are higher than skin threshold.</p> <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>	Format:	U5	Value	Name	[1,31]		26
Format:	U5								
Value	Name								
[1,31]									
26	[Default]								
1	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO								
Format:	MBZ								
0	<b>ACE Enable</b> <table border="1"> <tr> <td>Format:</td> <td>Enable</td> </tr> </table>	Format:	Enable						
Format:	Enable								
1	31:24	<b>Y3</b> <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 y_pixel for point 3 in PWL.</p>	Default Value:	76	Format:	U8			
	Default Value:	76							
	Format:	U8							
	23:16	<b>Y2</b> <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 y_pixel for point 2 in PWL.</p>	Default Value:	56	Format:	U8			
Default Value:	56								
Format:	U8								
15:8	<b>Y1</b> <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 y_pixel for point 1 in PWL.</p>	Default Value:	36	Format:	U8				
Default Value:	36								
Format:	U8								
7:0	<b>Ymin</b> <table border="1"> <tr> <td>Default Value:</td> <td>16</td> </tr> <tr> <td>Format:</td> <td>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	<b>Y7</b> <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 y_pixel for point 7 in PWL.</p>	Default Value:	156	Format:	U8			
		Default Value:	156						
Format:	U8								



<b>VEBOX_ACE_LACE_STATE</b>						
	23:16	<b>Y6</b> <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	<b>Y5</b> <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	<b>Y4</b> <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	<b>Ymax</b> <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	<b>Y10</b> <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	<b>Y9</b> <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	<b>Y8</b> <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					
4	31:24	<b>B4</b> <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 bias for point 4 in PWL.</p>	Default Value:	96	Format:	U8
		Default Value:	96			
Format:	U8					

<b>VEBOX_ACE_LACE_STATE</b>		
	23:16	<b>B3</b>
		Default Value: 76
		Format: U8 The value of the bias for point 3 in PWL.
	15:8	<b>B2</b>
		Default Value: 56
		Format: U8 The value of the bias for point 2 in PWL.
	7:0	<b>B1</b>
		Default Value: 36
		Format: U8 The value of the bias for point 1 in PWL.
5	31:24	<b>B8</b>
		Default Value: 176
		Format: U8 The value of the bias for point 8 in PWL.
	23:16	<b>B7</b>
		Default Value: 156
		Format: U8 The value of the bias for point 7 in PWL.
	15:8	<b>B6</b>
		Default Value: 136
		Format: U8 The value of the bias for point 6 in PWL.
	7:0	<b>B5</b>
		Default Value: 116
		Format: U8 The value of the bias for point 5 in PWL.
6	31:16	<b>Reserved</b>
		Access: RO
		Format: MBZ

<b>VEBOX_ACE_LACE_STATE</b>						
	15:8	<b>B10</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">216</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">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					
	7:0	<b>B9</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">196</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U8</td> </tr> </table> <p>The value of the bias for point 9 in PWL.</p>	Default Value:	196	Format:	U8
Default Value:	196					
Format:	U8					
7	31:27	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: right;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	26:16	<b>S1</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">1024</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U1.10</td> </tr> </table> <p>The value of the slope for point 1 in PWL</p> <p>The default is 1024/1024</p>	Default Value:	1024	Format:	U1.10
	Default Value:	1024				
Format:	U1.10					
15:11	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: right;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
10:0	<b>S0</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">1024</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U1.10</td> </tr> </table> <p>The value of the slope for point 0 in PWL</p> <p>The default is 1024/1024</p>	Default Value:	1024	Format:	U1.10	
Default Value:	1024					
Format:	U1.10					
8	31:27	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: right;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
26:16	<b>S3</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: right;">1024</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">U1.10</td> </tr> </table> <p>The value of the slope for point 3 in PWL</p> <p>The default is 1024/1024</p>	Default Value:	1024	Format:	U1.10	
Default Value:	1024					
Format:	U1.10					

## VEBOX\_ACE\_LACE\_STATE

	15:11	<b>Reserved</b>	Access:	RO	
			Format:	MBZ	
	10:0	<b>S2</b>	Default Value:	1024	
			Format:	U1.10	
			The value of the slope for point 2 in PWL The default is 1024/1024		
9	31:27	<b>Reserved</b>	Access:	RO	
			Format:	MBZ	
	26:16	<b>S5</b>	Default Value:	1024	
			Format:	U1.10	
			The value of the slope for point 5 in PWL The default is 1024/1024		
	15:11	<b>Reserved</b>	Access:	RO	
			Format:	MBZ	
	10:0	<b>S4</b>	Default Value:	1024	
			Format:	U1.10	
			The value of the slope for point 4 in PWL The default is 1024/1024		
	10	31:27	<b>Reserved</b>	Access:	RO
				Format:	MBZ
26:16		<b>S7</b>	Default Value:	1024	
			Format:	U1.10	
			The value of the slope for point 7 in PWL The default is 1024/1024		

<b>VEBOX_ ACE_LACE_STATE</b>		
	15:11	<b>Reserved</b>
		Access: RO Format: MBZ
	10:0	<b>S6</b>
		Default Value: 1024 Format: U1.10
The default is 1024/1024		
11	31:27	<b>Reserved</b>
		Access: RO Format: MBZ
	26:16	<b>S9</b>
		Default Value: 1024 Format: U1.10
		The value of the slope for point 9 in PWL The default is 1024/1024
	15:11	<b>Reserved</b>
		Access: RO Format: MBZ
	10:0	<b>S8</b>
		Default Value: 1024 Format: U1.10
		The value of the slope for point 8 in PWL The default is 1024/1024
12	31:16	<b>Max_ ACE_luma</b>
		Format: U16 The maximum luma for which ACE correction will be used.
	15:11	<b>Reserved</b>
		Access: RO Format: MBZ
	10:0	<b>S10</b>
Default Value: 1024 Format: U1.10		
The value of the slope for point 10 in PWL.		

<b>VEBOX_ ACE_LACE_STATE</b>					
13	31	<b>LACE Color Correction Enable</b> Enables LACE Color Correction			
	30:16	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				
15:0	<b>LACE Y Offset</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U16</td> </tr> </table> Y offset for LACE Color Correction	Format:	U16		
Format:	U16				
14	31:16	<b>LACE V Offset</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U16</td> </tr> </table> V offset for LACE Color Correction	Format:	U16	
	Format:	U16			
15:0	<b>LACE U Offset</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U16</td> </tr> </table> U offset for LACE Color Correction	Format:	U16		
Format:	U16				
15	31:21	<b>LACE gamma curve slope 0</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U1.10</td> </tr> </table> Slope value for segment 0	Format:	U1.10	
	Format:	U1.10			
	20:8	<b>LACE gamma curve bias 0</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U5.8</td> </tr> </table> Bias value for segment 0	Format:	U5.8	
Format:	U5.8				
7:0	<b>LACE gamma curve point 0</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U5.3</td> </tr> </table> Point value for segment 0 <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> This value must be 0	Format:	U5.3	<b>Programming Notes</b>	
Format:	U5.3				
<b>Programming Notes</b>					
16	31:21	<b>LACE gamma curve slope 1</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U1.10</td> </tr> </table> Slope value for segment 1	Format:	U1.10	
	Format:	U1.10			
	20:8	<b>LACE gamma curve bias 1</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U5.8</td> </tr> </table> Bias value for segment 1	Format:	U5.8	
Format:	U5.8				
7:0	<b>LACE gamma curve point 1</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td>U5.3</td> </tr> </table> Point value for segment 1	Format:	U5.3		
Format:	U5.3				

<b>VEBOX_ACE_LACE_STATE</b>		
17	31:21	<b>LACE gamma curve slope 2</b> Format: <span style="float: right;">U1.10</span> Slope value for segment 2
	20:8	<b>LACE gamma curve bias 2</b> Format: <span style="float: right;">U5.8</span> Bias value for segment 2
	7:0	<b>LACE gamma curve point 2</b> Format: <span style="float: right;">U5.3</span> Point value for segment 2
18	31:21	<b>LACE gamma curve slope 3</b> Format: <span style="float: right;">U1.10</span> Slope value for segment 3
	20:8	<b>LACE gamma curve bias 3</b> Format: <span style="float: right;">U5.8</span> Bias value for segment 3
	7:0	<b>LACE gamma curve point 3</b> Format: <span style="float: right;">U5.3</span> Point value for segment 3
19	31:21	<b>LACE gamma curve slope 4</b> Format: <span style="float: right;">U1.10</span> Slope value for segment 4
	20:8	<b>LACE gamma curve bias 4</b> Format: <span style="float: right;">U5.8</span> Bias value for segment 4
	7:0	<b>LACE gamma curve point 4</b> Format: <span style="float: right;">U5.3</span> Point value for segment 4
20	31:21	<b>LACE gamma curve slope 5</b> Format: <span style="float: right;">U1.10</span> Slope value for segment 5

<b>VEBOX_ACE_LACE_STATE</b>				
	20:8	<b>LACE gamma curve bias 5</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U5.8</td></tr></table> Bias value for segment 5		U5.8
		U5.8		
7:0	<b>LACE gamma curve point 5</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U5.3</td></tr></table> Point value for segment 5		U5.3	
	U5.3			
21	31:21	<b>LACE gamma curve slope 6</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U1.10</td></tr></table> Slope value for segment 6		U1.10
		U1.10		
	20:8	<b>LACE gamma curve bias 6</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U5.8</td></tr></table> Bias value for segment 6		U5.8
	U5.8			
7:0	<b>LACE gamma curve point 6</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U5.3</td></tr></table> Point value for segment 6		U5.3	
	U5.3			
22	31:21	<b>LACE gamma curve slope 7</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U1.10</td></tr></table> Slope value for segment 7		U1.10
		U1.10		
	20:8	<b>LACE gamma curve bias 7</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U5.8</td></tr></table> Bias value for segment 7		U5.8
	U5.8			
7:0	<b>LACE gamma curve point 7</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U5.3</td></tr></table> Point value for segment 7		U5.3	
	U5.3			
23	31:21	<b>LACE gamma curve slope 8</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U1.10</td></tr></table> Slope value for segment 8		U1.10
		U1.10		
20:8	<b>LACE gamma curve bias 8</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>U5.8</td></tr></table> Bias value for segment 8		U5.8	
	U5.8			



<b>VEBOX_ACE_LACE_STATE</b>		
24	7:0	<b>LACE gamma curve point 8</b> Format: U5.3 Point value for segment 8
	31:21	<b>LACE gamma curve slope 9</b> Format: U1.10 Slope value for segment 9
	20:8	<b>LACE gamma curve bias 9</b> Format: U5.8 Bias value for segment 9
25	7:0	<b>LACE gamma curve point 9</b> Format: U5.3 Point value for segment 9
	31:21	<b>LACE gamma curve slope 10</b> Format: U1.10 Slope value for segment 10
	20:8	<b>LACE gamma curve bias 10</b> Format: U5.8 Bias value for segment 10
26	7:0	<b>LACE gamma curve point 10</b> Format: U5.3 Point value for segment 10
	31:21	<b>LACE gamma curve slope 11</b> Format: U1.10 Slope value for segment 11
	20:8	<b>LACE gamma curve bias 11</b> Format: U5.8 Bias value for segment 11
26	7:0	<b>LACE gamma curve point 11</b> Format: U5.3 Point value for segment 11

<b>VEBOX_ACE_LACE_STATE</b>		
27	31:21	<b>LACE gamma curve slope 12</b> Format: <span style="float: right;">U1.10</span> Slope value for segment 12
		<b>LACE gamma curve bias 12</b> Format: <span style="float: right;">U5.8</span> Bias value for segment 12
	7:0	<b>LACE gamma curve point 12</b> Format: <span style="float: right;">U5.3</span> Point value for segment 12
28	31:21	<b>LACE gamma curve slope 13</b> Format: <span style="float: right;">U1.10</span> Slope value for segment 13
		<b>LACE gamma curve bias 13</b> Format: <span style="float: right;">U5.8</span> Bias value for segment 13
	7:0	<b>LACE gamma curve point 13</b> Format: <span style="float: right;">U5.3</span> Point value for segment 13
29	31:21	<b>LACE gamma curve slope 14</b> Format: <span style="float: right;">U1.10</span> Slope value for segment 14
		<b>LACE gamma curve bias 14</b> Format: <span style="float: right;">U5.8</span> Bias value for segment 14
	7:0	<b>LACE gamma curve point 14</b> Format: <span style="float: right;">U5.3</span> Point value for segment 14
30	31:21	<b>LACE gamma curve slope 15</b> Format: <span style="float: right;">U1.10</span> Slope value for segment 15

<b>VEBOX_ACE_LACE_STATE</b>		
	20:8	<b>LACE gamma curve bias 15</b> Format: U5.8 Bias value for segment 15
	7:0	<b>LACE gamma curve point 15</b> Format: U5.3 Point value for segment 15 <div style="text-align: center; background-color: #e6f2ff;"><b>Programming Notes</b></div> This value must be 0xff



## VEBOX\_ALPHA\_AOI\_STATE

VEBOX_ALPHA_AOI_STATE			
Source:	VideoEnhancementCS		
Size (in bits):	96		
Default Value:	0x00000000, 0x00000000, 0x00000000		
This state structure contains the IECP State Table Contents for Fixed Alpha State and Area of Interest State.			
DWord	Bit	Description	
0	31:18	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	17	<b>Full Image Histogram</b>	
		Default Value:	0
		Format:	Enable
	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).		
	16	<b>Alpha from State Select</b>	
		Format:	U1
		<b>Value</b>	<b>Name</b>
0		alpha is taken from message	
1		alpha is taken from state	
<b>Programming Notes</b>			
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.			
15:0	<b>Color Pipe Alpha</b>		
	Format:	U16	
	<b>Programming Notes</b>		
The 8 MSB of this field will be used for output formats that have 8-bits of alpha.			
1	31:30	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	29:16	<b>AOI Max X</b>	
		Default Value:	0
		Format:	U14

<b>VEBOX_ALPHA_AOI_STATE</b>						
	<p>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.</p> <p>The Area of Interest applies to the RGB Histogram and the White/Gray point sums as well.</p> <p style="text-align: center;"><b>Programming Notes</b></p> <p><b>This value must be a multiple of 4 minus 1.</b></p>					
15:14	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
13:0	<p><b>AOI Min X</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U14</td> </tr> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p><b>This value must be a multiple of 4.</b></p>	Default Value:	0	Format:	U14	
Default Value:	0					
Format:	U14					
2	31:30	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	29:16	<p><b>AOI Max Y</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U14</td> </tr> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p><b>This value must be a multiple of 4 minus 1.</b></p>	Default Value:	0	Format:	U14
Default Value:	0					
Format:	U14					
15:14	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
13:0	<p><b>AOI Min Y</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>U14</td> </tr> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p><b>This value must be a multiple of 4.</b></p>	Default Value:	0	Format:	U14	
Default Value:	0					
Format:	U14					



## VEBOX\_CAPTURE\_PIPE\_STATE

VEBOX_CAPTURE_PIPE_STATE			
Source:	VideoEnhancementCS		
Size (in bits):	224		
Default Value:	0x8511FF23, 0xAA64AFAA, 0xE6FD4000, 0x00000000, 0x00000000, 0x00000000, 0x00000000		
This command contains variables for controlling Demosaic and the White Balance Statistics.			
DWord	Bit	Description	
0	31:30	<b>DirMap_Scale</b>	
		Default Value:	2
		Format:	U2
	29:24	<b>Good Pixel Threshold</b>	
		Default Value:	5h
		Format:	U6
	The difference threshold between adjacent pixels for a pixel to be considered "good".		
	23	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	22:20	<b>Shift Min Cost</b>	
		Default Value:	1h
		Format:	U3
	The amount to shift the H2/V2 versions of min_cost.		
19:16	<b>Green Imbalance Threshold</b>		
	Default Value:	1h	
	Format:	U4	
15:8	<b>Average Color Threshold</b>		
	Default Value:	FFh	
	Format:	U8	
	The threshold between two colors in a pixel for the Avg interpolation to be considered.		
	<b>Programming Notes</b>		
Must be set to 255.			
7:6	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	

<b>VEBOX_CAPTURE_PIPE_STATE</b>						
	5:0	<b>Good Pixel Neighbor Threshold</b>				
		<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	<b>Scale For Min Cost</b>				
		<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	<b>Good Intesity Threshold</b>				
		<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	<b>Bad Color Threshold 1</b>					
	<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	<b>Bad Color Threshold 2</b>					
	<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	<b>Number Big Pixel Threshold</b>					
		<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	<b>Bad Color Threshold 3</b>					
	<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	<b>Y Bright Value</b>				
		<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				
	<table border="1"> <tr> <td colspan="2" style="text-align: center;"><b>Programming Notes</b></td> </tr> <tr> <td colspan="2">"00000000" is appended to the LSBs before comparing with Y.</td> </tr> </table>	<b>Programming Notes</b>		"00000000" is appended to the LSBs before comparing with Y.		
<b>Programming Notes</b>						
"00000000" is appended to the LSBs before comparing with Y.						

## VEBOX\_CAPTURE\_PIPE\_STATE

3	23:16	<b>Y Outlier Value</b> <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;"><b>Programming Notes</b></td> </tr> </table> <p>"00000000" is appended to the LSBs before comparing with Y.</p>	Default Value:	FDh	<b>Programming Notes</b>						
	Default Value:	FDh									
	<b>Programming Notes</b>										
	15:8	<b>UV Threshold Value</b> The value denotes the maximum threshold of the ratio between U+V to Y can have to be considered a gray point. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr style="background-color: #e6f2ff;"> <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><b>[Default]</b></td> <td><math>0.25 * 255 = 64</math></td> </tr> </tbody> </table>	Value	Name	Description	[255,0]		Encode a value from 255/256 to 0/256	64	<b>[Default]</b>	$0.25 * 255 = 64$
	Value	Name	Description								
	[255,0]		Encode a value from 255/256 to 0/256								
	64	<b>[Default]</b>	$0.25 * 255 = 64$								
	7	<b>Black Point Offset Red MSB</b>									
	6	<b>Black Point Offset Green Top MSB</b>									
	5	<b>Black Point Offset Blue MSB</b>									
	4	<b>Black Point Offset Green Bottom MSB</b>									
	3	<b>RGB Histogram Enable</b> Enables the collection of RGB Histograms for Auto-white balance correction and other uses. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr style="background-color: #e6f2ff;"> <th style="text-align: center;"><b>Programming Notes</b></th> </tr> </thead> <tbody> <tr> <td>This bit can be set without White Balance enable being set.</td> </tr> </tbody> </table>	<b>Programming Notes</b>	This bit can be set without White Balance enable being set.							
	<b>Programming Notes</b>										
This bit can be set without White Balance enable being set.											
2	<b>Vignette Correction Format</b> Defines what shift should be assumed for the <b>Vignette</b> Correction input values: <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr style="background-color: #e6f2ff;"> <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	<b>Black Point Correction Enable</b> <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table>	Format:	Enable								
Format:	Enable										
0	<b>White Balance Correction Enable</b> <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">Enable</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr style="background-color: #e6f2ff;"> <th style="text-align: center;"><b>Programming Notes</b></th> </tr> </thead> <tbody> <tr> <td>RGB Histogram enable must be set if this bit is set.</td> </tr> </tbody> </table>	Format:	Enable	<b>Programming Notes</b>	RGB Histogram enable must be set if this bit is set.						
Format:	Enable										
<b>Programming Notes</b>											
RGB Histogram enable must be set if this bit is set.											
31:16	<b>Black Point Offset Red</b> <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td style="width: 70%;">Default Value:</td> <td style="width: 30%;">0</td> </tr> <tr> <td>Format:</td> <td>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										



<b>VEBOX_CAPTURE_PIPE_STATE</b>						
	15:0	<p><b>Black Point Offset Green Top</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>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:	U16
Default Value:	0					
Format:	U16					
4	31:16	<p><b>Black Point Offset Blue</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>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:	U16
	Default Value:	0				
Format:	U16					
	15:0	<p><b>Black Point Offset Green Bottom</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>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:	U16
Default Value:	0					
Format:	U16					
5	31:16	<p><b>White Balance Red Correction</b></p> <table border="1"> <tr> <td>Format:</td> <td>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	<p><b>White Balance Green Top Correction</b></p> <table border="1"> <tr> <td>Format:</td> <td>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	<p><b>White Balance Blue Correction</b></p> <table border="1"> <tr> <td>Format:</td> <td>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	<p><b>White Balance Green Bottom Correction</b></p> <table border="1"> <tr> <td>Format:</td> <td>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

<b>VEBOX_CCM_STATE</b>		
Source:	VideoEnhancementCS	
Size (in bits):	480	
Default Value:	0x00004750, 0x0000AE80, 0x00000470, 0x00000220, 0x01FFFCC0, 0x0000D230, 0x00000A80, 0x01FFFF40, 0x0000D6A0, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
This state structure contains the IECP State Table Contents for the Color Correction Matrix State.		
DWord	Bit	Description
0	31	<b>Color Correction Matrix Enable</b>
		Format: Enable
		This bit enables the Color Correction Matrix.
		<b>Programming Notes</b>
	Single Pipe IECP Enable must also be set if this bit is enabled.	
	30:27	<b>Reserved</b>
Access: RO		
Format: MBZ		
26:0	<b>C1</b>	
	Default Value: 0004750h = 18256/65536	
	Format: S4.22	
	Coefficient of 3x3 Transform matrix. In HDR mode, bits [26:22] form the signed integer portion and bits [21:0] form the fraction portion of the coefficient. In non-HDR mode, bits [26:22] form the signed integer portion and bits [21:6] form the fraction portion of the coefficient.	
1	31:27	<b>Reserved</b>
		Access: RO
		Format: MBZ
	26:0	<b>C0</b>
		Default Value: 000AE80h = 44672/65536
		Format: S4.22
Coefficient of 3x3 Transform matrix. In HDR mode, bits [26:22] form the signed integer portion and bits [21:0] form the fraction portion of the coefficient. In non-HDR mode, bits [26:22] form the signed integer portion and bits [21:6] form the fraction portion of the coefficient.		
2	31:27	<b>Reserved</b>
		Access: RO
		Format: MBZ

<b>VEBOX_CCM_STATE</b>				
	26:0	<b>C3</b>		
		<table border="1"> <tr> <td>Default Value:</td> <td>0000470h = 1136/65536</td> </tr> <tr> <td>Format:</td> <td>S4.22</td> </tr> </table> <p>Coefficient of 3x3 Transform matrix. In HDR mode, bits [26:22] form the signed integer portion and bits [21:0] form the fraction portion of the coefficient. In non-HDR mode, bits [26:22] form the signed integer portion and bits [21:6] form the fraction portion of the coefficient.</p>	Default Value:	0000470h = 1136/65536
Default Value:	0000470h = 1136/65536			
Format:	S4.22			
3	31:27	<b>Reserved</b>		
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
	26:0	<b>C2</b>		
		<table border="1"> <tr> <td>Default Value:</td> <td>0000220h = 544/65536</td> </tr> <tr> <td>Format:</td> <td>S4.22</td> </tr> </table> <p>Coefficient of 3x3 Transform matrix. In HDR mode, bits [26:22] form the signed integer portion and bits [21:0] form the fraction portion of the coefficient. In non-HDR mode, bits [26:22] form the signed integer portion and bits [21:6] form the fraction portion of the coefficient.</p>	Default Value:	0000220h = 544/65536
Default Value:	0000220h = 544/65536			
Format:	S4.22			
4	31:27	<b>Reserved</b>		
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
	26:0	<b>C5</b>		
		<table border="1"> <tr> <td>Default Value:</td> <td>1FFCC0h = -832/65536</td> </tr> <tr> <td>Format:</td> <td>S4.22</td> </tr> </table> <p>Coefficient of 3x3 Transform matrix. In HDR mode, bits [26:22] form the signed integer portion and bits [21:0] form the fraction portion of the coefficient. In non-HDR mode, bits [26:22] form the signed integer portion and bits [21:6] form the fraction portion of the coefficient.</p>	Default Value:	1FFCC0h = -832/65536
Default Value:	1FFCC0h = -832/65536			
Format:	S4.22			
5	31:27	<b>Reserved</b>		
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
	26:0	<b>C4</b>		
		<table border="1"> <tr> <td>Default Value:</td> <td>000D230h = 53808/65536</td> </tr> <tr> <td>Format:</td> <td>S4.22</td> </tr> </table> <p>Coefficient of 3x3 Transform matrix. In HDR mode, bits [26:22] form the signed integer portion and bits [21:0] form the fraction portion of the coefficient. In non-HDR mode, bits [26:22] form the signed integer portion and bits [21:6] form the fraction portion of the coefficient.</p>	Default Value:	000D230h = 53808/65536
Default Value:	000D230h = 53808/65536			
Format:	S4.22			
6	31:27	<b>Reserved</b>		
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			

## VEBOX\_CCM\_STATE

	26:0	<b>C7</b>	
		Default Value:	0000A80h = 2688/65536
		Format:	S4.22
		Coefficient of 3x3 Transform matrix. In HDR mode, bits [26:22] form the signed integer portion and bits [21:0] form the fraction portion of the coefficient. In non-HDR mode, bits [26:22] form the signed integer portion and bits [21:6] form the fraction portion of the coefficient.	
7	31:27	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	26:0	<b>C6</b>	
		Default Value:	1FFFF40h = -192/65536
		Format:	S4.22
		Coefficient of 3x3 Transform matrix. In HDR mode, bits [26:22] form the signed integer portion and bits [21:0] form the fraction portion of the coefficient. In non-HDR mode, bits [26:22] form the signed integer portion and bits [21:6] form the fraction portion of the coefficient.	
8	31:27	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	26:0	<b>C8</b>	
		Default Value:	000D6A0h = 54944/65536
		Format:	S4.22
		Coefficient of 3x3 Transform matrix. In HDR mode, bits [26:22] form the signed integer portion and bits [21:0] form the fraction portion of the coefficient. In non-HDR mode, bits [26:22] form the signed integer portion and bits [21:6] form the fraction portion of the coefficient.	
9	31:0	<b>Offset_in_R</b>	
		Default Value:	0
		Format:	S31
		The input offset for red component. In HDR mode, the range of the value is $-2^{31}$ to $2^{31}-1$ . In non-HDR mode, the range of the value is $-2^{16}$ to $2^{16}-1$	
10	31:0	<b>Offset_in_G</b>	
		Default Value:	0
		Format:	S31
		The input offset for green component. In HDR mode, the range of the value is $-2^{31}$ to $2^{31}-1$ . In non-HDR mode, the range of the value is $-2^{16}$ to $2^{16}-1$	

<b>VEBOX_CCM_STATE</b>		
11	31:0	<b>Offset_in_B</b>
		Default Value: 0
		Format: S31
		The input offset for blue component. In HDR mode, the range of the value is $-2^{31}$ to $2^{31}-1$ . In non-HDR mode, the range of the value is $-2^{16}$ to $2^{16}-1$
12	31:0	<b>Offset_out_R</b>
		Default Value: 0
		Format: S31
		The output offset for red component. In HDR mode, the range of the value is $-2^{31}$ to $2^{31}-1$ . In non-HDR mode, the range of the value is $-2^{16}$ to $2^{16}-1$
13	31:0	<b>Offset_out_G</b>
		Default Value: 0
		Format: S31
		The output offset for green component. In HDR mode, the range of the value is $-2^{31}$ to $2^{31}-1$ . In non-HDR mode, the range of the value is $-2^{16}$ to $2^{16}-1$
14	31:0	<b>Offset_out_B</b>
		Default Value: 0
		Format: S31
		The output offset for blue component. In HDR mode, the range of the value is $-2^{31}$ to $2^{31}-1$ . In non-HDR mode, the range of the value is $-2^{16}$ to $2^{16}-1$



## VEBOX\_Ch\_Dir\_Filter\_Coefficient

VEBOX_Ch_Dir_Filter_Coefficient				
Size (in bits):	64			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0..1	63:56	<b>Filter Coefficient[7]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6
		S1.6		
	55:48	<b>Filter Coefficient[6]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6
		S1.6		
	47:40	<b>Filter Coefficient[5]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6
		S1.6		
	39:32	<b>Filter Coefficient[4]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6
		S1.6		
31:24	<b>Filter Coefficient[3]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6	
	S1.6			
23:16	<b>Filter Coefficient[2]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6	
	S1.6			
15:8	<b>Filter Coefficient[1]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6	
	S1.6			
7:0	<b>Filter Coefficient[0]</b> Format: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td>S1.6</td></tr></table> <b>Range:</b> [-2, +2)		S1.6	
	S1.6			

## 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	<b>Transform Enable</b> Format: Enable
	30:19	<b>Reserved</b> Access: RO Format: MBZ
		18:0
1	31:19	<b>Reserved</b> Access: RO Format: MBZ
	18:0	<b>C1</b> Default Value: 0 Format: S2.16 Transform coefficient.
2	31:19	<b>Reserved</b> Access: RO Format: MBZ
	18:0	<b>C2</b> Default Value: 0 Format: S2.16 Transform coefficient.
3	31:19	<b>Reserved</b> Access: RO Format: MBZ

VEBOX_CSC_STATE						
	18:0	<b>C3</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S2.16</td> </tr> </table> Transform coefficient.	Default Value:	0	Format:	S2.16
Default Value:	0					
Format:	S2.16					
4	31:19	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
	18:0	<b>C4</b> <table border="1"> <tr> <td>Default Value:</td> <td>10000h or 1.0</td> </tr> <tr> <td>Format:</td> <td>S2.16</td> </tr> </table> Transform coefficient.	Default Value:	10000h or 1.0	Format:	S2.16
Default Value:	10000h or 1.0					
Format:	S2.16					
5	31:19	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
	18:0	<b>C5</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S2.16</td> </tr> </table> Transform coefficient.	Default Value:	0	Format:	S2.16
Default Value:	0					
Format:	S2.16					
6	31:19	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
	18:0	<b>C6</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S2.16</td> </tr> </table> Transform coefficient.	Default Value:	0	Format:	S2.16
Default Value:	0					
Format:	S2.16					
7	31:19	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
	18:0	<b>C7</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S2.16</td> </tr> </table> Transform coefficient.	Default Value:	0	Format:	S2.16
Default Value:	0					
Format:	S2.16					



<b>VEBOX_CSC_STATE</b>						
8	31:19	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
18:0	<b>C8</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>10000h or 1.0</td> </tr> <tr> <td>Format:</td> <td>S2.16</td> </tr> </table> Transform coefficient. The offset value is multiplied by 2 before being added to the output.	Default Value:	10000h or 1.0	Format:	S2.16	
Default Value:	10000h or 1.0					
Format:	S2.16					
9	31:16	<b>Offset Out 1</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15</td> </tr> </table> Offset in for Y/R. The offset value is multiplied by 2 before being added to the output.	Default Value:	0	Format:	S15
	Default Value:	0				
Format:	S15					
15:0	<b>Offset in 1</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15</td> </tr> </table> Offset in for Y/R. The offset value is multiplied by 2 before being added to the output.	Default Value:	0	Format:	S15	
Default Value:	0					
Format:	S15					
10	31:16	<b>Offset Out 2</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15</td> </tr> </table> Offset out for U/G. The offset value is multiplied by 2 before being added to the output.	Default Value:	0	Format:	S15
	Default Value:	0				
Format:	S15					
15:0	<b>Offset in 2</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15</td> </tr> </table> Offset out for U/G. The offset value is multiplied by 2 before being added to the output.	Default Value:	0	Format:	S15	
Default Value:	0					
Format:	S15					
11	31:16	<b>Offset Out 3</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15</td> </tr> </table> Offset out for V/B. The offset value is multiplied by 2 before being added to the output.	Default Value:	0	Format:	S15
	Default Value:	0				
Format:	S15					
15:0	<b>Offset in 3</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S15</td> </tr> </table> Offset out for V/B. The offset value is multiplied by 2 before being added to the output.	Default Value:	0	Format:	S15	
Default Value:	0					
Format:	S15					



## VEBOX\_DNDI\_STATE

<b>VEBOX_DNDI_STATE</b>											
Source:	VideoEnhancementCS										
Size (in bits):	1568										
Default Value:	0x00000000, 0x80000000, 0x00000400, 0x00000000, 0x00000000, 0x00000000, 0x000000A0, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x0800F000, 0x0F025800, 0x00007800, 0x2D000000, 0x00000254, 0x00000000, 0x00000400, 0x05FF05FF, 0x00000000, 0x005064A5, 0x00000000, 0x00000000, 0x1314640F, 0x00000000, 0xFF804020, 0x5DC48F64, 0x190F8DA4, 0x03FFFFFF, 0x00043CA0										
This state table is used by the <i>Denoise and Deinterlacer functions</i>											
DWord	Bit	Description									
0	31:17	<b>Denoise STAD Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U15</td> </tr> </table> Threshold for denoise sum of temporal absolute differences.	Format:	U15							
	Format:	U15									
	16:8	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:		RO									
Format:	MBZ										
7:0	<b>Denoise Maximum History</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> Maximum allowed value for denoise history. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 60%;">Value</th> <th style="width: 40%;">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]											
1	31:28	<b>Denoise History increase</b> Amount that denoise_history is increased by. MAX:15 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 45%;">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>	Value	Name	Description	8h	[Default]		15		Maximum Allowed
		Value	Name	Description							
		8h	[Default]								
	15		Maximum Allowed								
27:23	<b>Denoise Moving Pixel Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U5</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 60%;">Value</th> <th style="width: 40%;">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]											

<b>VEBOX_DNDI_STATE</b>			
	22:12	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	11:0	<b>Denoise ASD Threshold</b>	
	Format: U12	Threshold for denoise absolute sum of differences.	
		<b>Value</b>	<b>Name</b>
		[0,1023]	
2	31:20	<b>Temporal Difference Threshold</b>	
		Format: U12	
		<b>Programming Notes</b>	
		0 < (Temporal Difference Threshold - Low Temporal Difference Threshold) <= 256 except when both thresholds are set to 0.	
	19:11	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	10:5	<b>Initial Denoise History</b>	
		Default Value: 32	
		Format: U6	
	<b>Programming Notes</b>		
	Initial value for Denoise history for both Luma and Chroma		
4:0	<b>Reserved</b>		
	Access: RO		
	Format: MBZ		
3	31:20	<b>Low Temporal Difference Threshold</b>	
		Format: U12	
		0 < (Temporal Difference Threshold - Low Temporal Difference Threshold) <= 256 except when both thresholds are set to 0.	
	19:12	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
11	<b>Reserved</b>		
	Access: RO		
	Format: MBZ		

## VEBOX\_DNDI\_STATE

	10	<b>Progressive DN</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>Indicates that the denoise algorithm should assume progressive input when filtering neighboring pixels. <b>This bit must be set if the input to Denoise is RGB.</b></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></td> <td>DN assumes interlaced video and filters alternate lines together</td> </tr> <tr> <td>1</td> <td></td> <td>DN assumes progressive video and filters neighboring lines together</td> </tr> </tbody> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td><b>DI Enable</b> must be disabled when this field is enabled.</td> </tr> </table>	Format:	Enable	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	<b>DI Enable</b> must be disabled when this field is enabled.
	Format:	Enable													
	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															
<b>DI Enable</b> must be disabled when this field is enabled.															
9:2	<b>Hot Pixel Count Luma</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U8</td> </tr> </table> <p>Number of neighboring pixels different more than <b>Hot Pixel Threshold</b> before a pixel is considered hot.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 40%;">Value</th> <th style="width: 60%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,8]</td> <td></td> </tr> </tbody> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <th style="text-align: center;">Programming Notes</th> </tr> <tr> <td>0 will cause all pixels to be considered hot and will perform a median filter on the entire image.</td> </tr> </table>	Format:	U8	Value	Name	[0,8]		Programming Notes	0 will cause all pixels to be considered hot and will perform a median filter on the entire image.						
Format:	U8														
Value	Name														
[0,8]															
Programming Notes															
0 will cause all pixels to be considered hot and will perform a median filter on the entire image.															
1:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ										
Access:	RO														
Format:	MBZ														
4	<b>31:20 Denoise Threshold for Sum of Complexity Measure Luma</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U12</td> </tr> </table>	Format:	U12												
Format:	U12														
	<b>19:12 Hot Pixel Threshold Luma</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>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														
	<b>11:0 Block Noise Estimate Noise Threshold</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U12</td> </tr> </table> <p>Threshold for noise maximum/minimum.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 40%;">Value</th> <th style="width: 60%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,4095]</td> <td></td> </tr> </tbody> </table>	Format:	U12	Value	Name	[0,4095]									
Format:	U12														
Value	Name														
[0,4095]															
5	<b>31:17 Chroma Denoise STAD Threshold</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U15</td> </tr> </table> <p>Threshold for denoise sum of temporal absolute differences.</p>	Format:	U15												
Format:	U15														

<b>VEBOX_DNDI_STATE</b>												
	16	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
	Access:	RO										
	Format:	MBZ										
15:8	<b>Hot Pixel Threshold Chroma U</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>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											
7:0	<b>Hot Pixel Count Chroma U</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U8</td> </tr> </table> <p>Number of neighboring pixels different more than <b>Hot Pixel Threshold</b> before a pixel is considered hot</p>	Format:	U8									
Format:	U8											
6	31:20	<b>Chroma Temporal Difference Threshold</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U12</td> </tr> </table> <p><math>0 &lt; (\text{Chroma Temporal Difference Threshold} - \text{Chroma Low Temporal Difference Threshold}) \leq 256</math> except when both thresholds are set to 0</p>	Format:	U12								
	Format:	U12										
	19:12	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
	Access:	RO										
Format:	MBZ											
11:1	<b>Block Noise Estimate Edge Threshold</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Default Value:</td> <td>80</td> </tr> </table> <p>Threshold for detecting an edge in block noise estimate.</p>	Default Value:	80									
Default Value:	80											
0	<b>Chroma Denoise Enable</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">Value</th> <th style="width: 15%;">Name</th> <th style="width: 75%;">Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>The U and V channels will be passed to the next stage after DN unchanged.</td> </tr> <tr> <td>1</td> <td></td> <td>The U and V chroma channels will be denoise filtered.</td> </tr> </tbody> </table>	Format:	Enable	Value	Name	Description	0		The U and V channels will be passed to the next stage after DN unchanged.	1		The U and V chroma channels will be denoise filtered.
Format:	Enable											
Value	Name	Description										
0		The U and V channels will be passed to the next stage after DN unchanged.										
1		The U and V chroma channels will be denoise filtered.										
7	31:20	<b>Chroma Low Temporal Difference Threshold</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U12</td> </tr> </table> <p><math>0 &lt; (\text{Chroma Temporal Difference Threshold} - \text{Chroma Low Temporal Difference Threshold}) \leq 256</math> except when both thresholds are set to 0</p>	Format:	U12								
	Format:	U12										
19:16	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ							
Access:	RO											
Format:	MBZ											

<b>VEBOX_DNDI_STATE</b>						
	15:8	<b>Hot Pixel Threshold Chroma V</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>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				
7:0	<b>Hot Pixel Count Chroma V</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Number of neighboring pixels different more than <b>Hot Pixel Threshold</b> before a pixel is considered hot</p>	Format:	U8			
Format:	U8					
8	31:29	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	28:24	<b>Chroma Denoise Moving Pixel Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U5</td> </tr> </table>	Format:	U5		
	Format:	U5				
23:12	<b>Chroma Denoise ASD Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U12</td> </tr> </table> <p>Threshold for denoise absolute sum of differences.</p>	Format:	U12			
Format:	U12					
11:0	<b>Chroma Denoise Threshold for Sum of Complexity Measure</b>					
9	31:30	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	29:25	<b>DnY_Wr5[4:0]</b> Weight to be applied when: $th4 \leq$ (difference in luma, Bayer or RGB value)				
	24:20	<b>DnY_Wr4[4:0]</b> Weight to be applied when: $th3 \leq$ (difference in luma, Bayer or RGB value) < $th4$				
	19:15	<b>DnY_Wr3[4:0]</b> Weight to be applied when: $th2 \leq$ (difference in luma, Bayer or RGB value) < $th3$				
	14:10	<b>DnY_Wr2[4:0]</b> Weight to be applied when: $th1 \leq$ (difference in luma, Bayer or RGBvalue) < $th2$				
	9:5	<b>DnY_Wr1[4:0]</b> Weight to be applied when: $th0 \leq$ (difference in luma, Bayer or RGB value) < $th1$				
4:0	<b>DnY_Wr0[4:0]</b> Weight to be applied when: (difference in luma, Bayer or RGB value) < $th0$					
10	31:29	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					

<b>VEBOX_DNDI_STATE</b>		
	28:16	<b>DnY_thmax[12:0]</b> Maximum threshold value for luma, Bayer or RGB
	15:13	<b>Reserved</b>
		Access: RO Format: MBZ
	12:0	<b>DnY_thmin[12:0]</b> Minimum threshold value
11	31:29	<b>Reserved</b>
		Access: RO Format: MBZ
	28:16	<b>DnY_prt5[12:0]</b>
	15:13	<b>Reserved</b>
		Access: RO Format: MBZ
		12:0
12	31:29	<b>Reserved</b>
		Access: RO Format: MBZ
	28:16	<b>DnY_prt4[12:0]</b> Multiplied by thrscale and then used as the threshold for comparing the luma or RGB differences.
	15:13	<b>Reserved</b>
		Access: RO Format: MBZ
	12:0	<b>DnY_prt3[12:0]</b>
13	31:29	<b>Reserved</b>
		Access: RO Format: MBZ
	28:16	<b>DnY_prt2[12:0]</b>
	15:13	<b>Reserved</b>
		Access: RO Format: MBZ
	12:0	<b>DnY_prt1[12:0]</b>
14	31:29	<b>Reserved</b>
		Access: RO Format: MBZ

<b>VEBOX_DNDI_STATE</b>			
	28:16	<b>DnY_prt0[12:0]</b>	
	15	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	14:10	<b>DnY_wd22[4:0]</b> Weight to be applied to the 4 luma, Bayer or RGB pixels that are at X2 and Y2	
	9:5	<b>DnY_wd21[4:0]</b> Weight to be applied to the 4 luma, Bayer or RGB pixels that are at X1 and Y2	
4:0	<b>DnY_wd20[4:0]</b> Weight to be applied to the 2 luma, Bayer or RGB pixels that are at X and Y2		
15	31:30	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	29:25	<b>DnY_wd12[4:0]</b> Weight to be applied to the 4 luma, Bayer or RGB pixels that are at X2 and Y1	
	24:20	<b>DnY_wd11[4:0]</b> Weight to be applied to the 4 luma, Bayer or RGB pixels that are at X1 and Y1	
	19:15	<b>DnY_wd10[4:0]</b> Weight to be applied to the 2 luma, Bayer or RGB pixels that are at X and Y1	
	14:10	<b>DnY_wd02[4:0]</b> Weight to be applied to the 2 luma, Bayer or RGB pixels that are at X2 and Y	
	9:5	<b>DnY_wd01[4:0]</b> Weight to be applied to the 2 luma, Bayer or RGB pixels that are at X1 and Y	
4:0	<b>DnY_wd00[4:0]</b> Weight to be applied to the 1 luma, Bayer or RGB pixels that are at X and Y		
16	31:30	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	29:25	<b>DnU_Wr5[4:0]</b> Weight to be applied when: $th4 \leq (\text{difference in chroma U value})$	
	24:20	<b>DnU_Wr4[4:0]</b> Weight to be applied when: $th3 \leq (\text{difference in chroma U value}) < th4$	
	19:15	<b>DnU_Wr3[4:0]</b> Weight to be applied when: $th2 \leq (\text{difference in chroma U value}) < th3$	
	14:10	<b>DnU_Wr2[4:0]</b> Weight to be applied when: $th1 \leq (\text{difference in chroma U value}) < th2$	
	9:5	<b>DnU_Wr1[4:0]</b> Weight to be applied when: $th0 \leq (\text{difference in chroma U value}) < th1$	
4:0	<b>DnU_Wr0[4:0]</b> Weight to be applied when: $(\text{difference in chroma U value}) < th0$		



<b>VEBOX_DNDI_STATE</b>		
17	31:29	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:16	<b>DnU_thmax[12:0]</b> Maximum threshold value for chroma U
	15:13	<b>Reserved</b>
Access: RO		
Format: MBZ		
12:0	<b>DnU_thmin[12:0]</b> Minimum threshold value	
18	31:29	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:16	<b>DnU_prt5[12:0]</b>
	15:13	<b>Reserved</b>
Access: RO		
Format: MBZ		
12:0	<b>DnU_dyn_thmin[12:0]</b> Minimum Dynamic threshold value.	
19	31:29	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:16	<b>DnU_prt4[12:0]</b> Multiplied by thrscale and then used as the threshold for comparing chroma U differences.
	15:13	<b>Reserved</b>
Access: RO		
Format: MBZ		
12:0	<b>DnU_prt3[12:0]</b>	
20	31:29	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:16	<b>DnU_prt2[12:0]</b>
	15:13	<b>Reserved</b>
Access: RO		
Format: MBZ		
12:0	<b>DnU_prt1[12:0]</b>	

<b>VEBOX_DNDI_STATE</b>		
21	31:29	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:16	<b>DnU_prt0[12:0]</b>
	15	<b>Reserved</b>
		Access: RO
		Format: MBZ
14:10	<b>DnU_wd22[4:0]</b> Weight to be applied to the 4 chroma U pixels that are at X2 and Y2	
9:5	<b>DnU_wd21[4:0]</b> Weight to be applied to the 4 chroma U pixels that are at X1 and Y2	
4:0	<b>DnU_wd20[4:0]</b> Weight to be applied to the 2 chroma U pixels that are at X and Y2	
22	31:30	<b>Reserved</b>
		Access: RO
		Format: MBZ
	29:25	<b>DnU_wd12[4:0]</b> Weight to be applied to the 4 chroma U pixels that are at X2 and Y1
	24:20	<b>DnU_wd11[4:0]</b> Weight to be applied to the 4 chroma U pixels that are at X1 and Y1
	19:15	<b>DnU_wd10[4:0]</b> Weight to be applied to the 2 chroma U pixels that are at X and Y1
	14:10	<b>DnU_wd02[4:0]</b> Weight to be applied to the 2 chroma U pixels that are at X2 and Y
	9:5	<b>DnU_wd01[4:0]</b> Weight to be applied to the 2 chroma U pixels that are at X1 and Y
4:0	<b>DnU_wd00[4:0]</b> Weight to be applied to the 1 chroma U pixels that are at X and Y	
23	31:30	<b>Reserved</b>
		Access: RO
		Format: MBZ
	29:25	<b>DnV_Wr5[4:0]</b> Weight to be applied when: $th4 \leq (\text{difference in chroma V value})$
	24:20	<b>DnV_Wr4[4:0]</b> Weight to be applied when: $th3 \leq (\text{difference in chroma V value}) < th4$
	19:15	<b>DnV_Wr3[4:0]</b> Weight to be applied when: $th2 \leq (\text{difference in chroma V value}) < th3$
14:10	<b>DnV_Wr2[4:0]</b> Weight to be applied when: $th1 \leq (\text{difference in chroma V value}) < th2$	

<b>VEBOX_DNDI_STATE</b>					
	9:5	<b>DnV_Wr51[4:0]</b> Weight to be applied when: $th0 \leq (\text{difference in chroma V value}) < th1$			
	4:0	<b>DnV_Wr0[4:0]</b> Weight to be applied when: $(\text{difference in chroma V value}) < th0$			
24	31:29	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
28:16	<b>DnV_thmax[12:0]</b> Maximum threshold value for chroma V				
15:13	<b>Reserved</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				
	12:0	<b>DnV_thmin[12:0]</b> Minimum threshold value			
25	31:29	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
28:16	<b>DnV_prt5[12:0]</b>				
15:13	<b>Reserved</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				
	12:0	<b>DnV_dyn_thmin[12:0]</b> Minimum Dynamic threshold value.			
26	31:29	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
	Format:	MBZ			
28:16	<b>DnV_prt4[12:0]</b> Multiplied by thrscale and then used as the threshold for comparing chroma V differences.				
15:13	<b>Reserved</b>				
	<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO				
Format:	MBZ				
	12:0	<b>DnV_prt3[12:0]</b>			
27	31:29	<b>Reserved</b>			
		<table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
	Access:	RO			
Format:	MBZ				
28:16	<b>DnV_prt2[12:0]</b>				
	15:13	<b>Reserved</b>			

<b>VEBOX_DNDI_STATE</b>		
		Access: RO
		Format: MBZ
	12:0	<b>DnV_prt1[12:0]</b>
28	31:29	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:16	<b>DnV_prt0[12:0]</b>
	15	<b>Reserved</b>
		Access: RO
		Format: MBZ
	14:10	<b>DnV_wd22[4:0]</b> Weight to be applied to the 4 chroma V pixels that are at X2 and Y2
	9:5	<b>DnV_wd21[4:0]</b> Weight to be applied to the 4 chroma V pixels that are at X1 and Y2
	4:0	<b>DnV_wd20[4:0]</b> Weight to be applied to the 2 chroma V pixels that are at X and Y2
29	31:30	<b>Reserved</b>
		Access: RO
		Format: MBZ
	29:25	<b>DnV_wd12[4:0]</b> Weight to be applied to the 4 chroma V pixels that are at X2 and Y1
	24:20	<b>DnV_wd11[4:0]</b> Weight to be applied to the 4 chroma V pixels that are at X1 and Y1
	19:15	<b>DnV_wd10[4:0]</b> Weight to be applied to the 2 chroma V pixels that are at X and Y1
	14:10	<b>DnV_wd02[4:0]</b> Weight to be applied to the 2 chroma V pixels that are at X2 and Y
	9:5	<b>DnV_wd01[4:0]</b> Weight to be applied to the 2 chroma V pixels that are at X1 and Y
30	31:17	<b>Eight Direction Edge Threshold</b>
		Default Value: 1024
		Format: U15
		Threshold to determine an edge in eight directional edge detector
	16:7	<b>Valid Pixel Threshold</b>
		Default Value: 480
		Format: U10

<b>VEBOX_DNDI_STATE</b>						
	6:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					
31	31:19	<b>Small Sobel Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>480</td> </tr> <tr> <td>Format:</td> <td>U13</td> </tr> </table> Threshold for weak Sobel response	Default Value:	480	Format:	U13
		Default Value:	480			
		Format:	U13			
<b>Large Sobel Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>2400</td> </tr> <tr> <td>Format:</td> <td>U13</td> </tr> </table> Threshold for strong Sobel response <div style="text-align: center; background-color: #e6f2ff; padding: 2px;"><b>Programming Notes</b></div> Large Sobel Threshold > Small Sobel Threshold	Default Value:	2400	Format:	U13		
Default Value:	2400					
Format:	U13					
5:0	<b>Small Sobel Count Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> Threshold for number of pixels in a block that have weak Sobel response (Default: 6)	Format:	U6			
Format:	U6					
32	31:26	<b>Median Sobel Count Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> Threshold for number of pixels in a block that have regular Sobel response (Default: 40)	Format:	U6		
	Format:	U6				
	25:20	<b>Large Sobel Count Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> Threshold for number of pixels in a block that have strong Sobel response (Default: 6)	Format:	U6		
	Format:	U6				
19:6	<b>Block Sigma Diff Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>480</td> </tr> <tr> <td>Format:</td> <td>U14</td> </tr> </table> Threshold for the difference between maximum and minimum sigma within a block	Default Value:	480	Format:	U14	
Default Value:	480					
Format:	U14					
5:0	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
33	31:19	<b>Max Sobel Threshold</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td>1440</td> </tr> <tr> <td>Format:</td> <td>U13</td> </tr> </table>	Default Value:	1440	Format:	U13
	Default Value:	1440				
Format:	U13					
18:0	<b>Reserved</b>					

<b>VEBOX_DNDI_STATE</b>															
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ									
Access:	RO														
Format:	MBZ														
34	31:16	<b>Reserved</b>													
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ									
	Access:	RO													
	Format:	MBZ													
	15:13	<b>Reserved</b>													
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ									
	Access:	RO													
	Format:	MBZ													
	12:10	<b>STMM C2</b>													
		<table border="1"> <tr> <td>Format:</td> <td>U3</td> </tr> </table> <p>Bias for divisor in STMM equation.</p> <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>	Format:	U3	Value	Name	Description	[0,7]		Representing values [1,8]					
Format:		U3													
Value		Name	Description												
[0,7]		Representing values [1,8]													
9:6	<b>Content Adaptive Threshold Slope</b>														
	<table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <p>Determines the slope of the Content Adaptive Threshold.</p> <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> <table border="1"> <thead> <tr> <th colspan="3">Programming Notes</th> </tr> </thead> <tbody> <tr> <td colspan="3">+1 added internally to get CAT_slope.</td> </tr> </tbody> </table>	Format:	U4	Value	Name	Description	9	[Default]	CAT_slope value = 10	Programming Notes			+1 added internally to get CAT_slope.		
	Format:	U4													
	Value	Name	Description												
9	[Default]	CAT_slope value = 10													
Programming Notes															
+1 added internally to get CAT_slope.															
5:2	<b>SAD Tight Threshold</b>														
	<table border="1"> <tr> <td>Default Value:</td> <td>5</td> </tr> <tr> <td>Format:</td> <td>U4</td> </tr> </table>	Default Value:	5	Format:	U4										
Default Value:	5														
Format:	U4														
1:0	<b>Smooth MV Threshold</b>														
	<table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table>	Format:	U2												
Format:	U2														
35	31	<b>STMM Blending Constant Select</b>													
		<table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></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>	Format:	U1	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		
		Format:	U1												
	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	<b>Blending constant across time for large values of STMM</b>														
23:16	<b>Blending constant across time for small values of STMM</b>														
	<table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table>	Format:	U8												
Format:	U8														
15:14	<b>Reserved</b>														

<b>VEBOX_DNDI_STATE</b>																																											
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ																																						
Access:	RO																																										
Format:	MBZ																																										
	<table border="1"> <tr> <td>13:8</td> <td> <b>Multiplier for VECM</b>  <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Determines the strength of the vertical edge complexity measure.</p> </td> </tr> <tr> <td>7:0</td> <td> <b>Maximum STMM</b>  <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Largest allowed STMM in blending equations.</p> </td> </tr> </table>	13:8	<b>Multiplier for VECM</b> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Determines the strength of the vertical edge complexity measure.</p>	Format:	U6	7:0	<b>Maximum STMM</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Largest allowed STMM in blending equations.</p>	Format:	U8																																		
13:8	<b>Multiplier for VECM</b> <table border="1"> <tr> <td>Format:</td> <td>U6</td> </tr> </table> <p>Determines the strength of the vertical edge complexity measure.</p>	Format:	U6																																								
Format:	U6																																										
7:0	<b>Maximum STMM</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Largest allowed STMM in blending equations.</p>	Format:	U8																																								
Format:	U8																																										
36	<table border="1"> <tr> <td>31:24</td> <td> <b>Minimum STMM</b>  <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Smallest allowed STMM in blending equations</p> </td> </tr> <tr> <td>23:22</td> <td> <b>STMM Shift Down</b>  <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Amount to shift STMM down (quantize to fewer bits)</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Shift by 4</td> </tr> <tr> <td>1</td> <td>Shift by 5</td> </tr> <tr> <td>2</td> <td>Shift by 6</td> </tr> <tr> <td>3</td> <td>Reserved</td> </tr> </tbody> </table> </td> </tr> <tr> <td>21:20</td> <td> <b>STMM Shift Up</b>  <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Amount to shift STMM up (set range).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Shift by 6</td> </tr> <tr> <td>1</td> <td>Shift by 7</td> </tr> <tr> <td>2</td> <td>Shift by 8</td> </tr> <tr> <td>3</td> <td>Reserved</td> </tr> </tbody> </table> </td> </tr> <tr> <td>19:16</td> <td> <b>STMM Output Shift</b>  <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,15]</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>The value of this field must satisfy the following equation: <math>stmm\_max - stmm\_min = 2^{\wedge}</math></td> </tr> </tbody> </table> </td> </tr> </table>	31:24	<b>Minimum STMM</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Smallest allowed STMM in blending equations</p>	Format:	U8	23:22	<b>STMM Shift Down</b> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Amount to shift STMM down (quantize to fewer bits)</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Shift by 4</td> </tr> <tr> <td>1</td> <td>Shift by 5</td> </tr> <tr> <td>2</td> <td>Shift by 6</td> </tr> <tr> <td>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	21:20	<b>STMM Shift Up</b> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Amount to shift STMM up (set range).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Shift by 6</td> </tr> <tr> <td>1</td> <td>Shift by 7</td> </tr> <tr> <td>2</td> <td>Shift by 8</td> </tr> <tr> <td>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	19:16	<b>STMM Output Shift</b> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,15]</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>The value of this field must satisfy the following equation: <math>stmm\_max - stmm\_min = 2^{\wedge}</math></td> </tr> </tbody> </table>	Format:	U4	Value	Name	[0,15]		Programming Notes	The value of this field must satisfy the following equation: $stmm\_max - stmm\_min = 2^{\wedge}$
31:24	<b>Minimum STMM</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Smallest allowed STMM in blending equations</p>	Format:	U8																																								
Format:	U8																																										
23:22	<b>STMM Shift Down</b> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Amount to shift STMM down (quantize to fewer bits)</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Shift by 4</td> </tr> <tr> <td>1</td> <td>Shift by 5</td> </tr> <tr> <td>2</td> <td>Shift by 6</td> </tr> <tr> <td>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	<b>STMM Shift Up</b> <table border="1"> <tr> <td>Format:</td> <td>U2</td> </tr> </table> <p>Amount to shift STMM up (set range).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Shift by 6</td> </tr> <tr> <td>1</td> <td>Shift by 7</td> </tr> <tr> <td>2</td> <td>Shift by 8</td> </tr> <tr> <td>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	<b>STMM Output Shift</b> <table border="1"> <tr> <td>Format:</td> <td>U4</td> </tr> </table> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>[0,15]</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Programming Notes</th> </tr> </thead> <tbody> <tr> <td>The value of this field must satisfy the following equation: <math>stmm\_max - stmm\_min = 2^{\wedge}</math></td> </tr> </tbody> </table>	Format:	U4	Value	Name	[0,15]		Programming Notes	The value of this field must satisfy the following equation: $stmm\_max - stmm\_min = 2^{\wedge}$																																		
Format:	U4																																										
Value	Name																																										
[0,15]																																											
Programming Notes																																											
The value of this field must satisfy the following equation: $stmm\_max - stmm\_min = 2^{\wedge}$																																											

<b>VEBOX_DNDI_STATE</b>			
		stmm_output_shift	
	15:12	<b>ChromaTDM_WT</b>	
		Default Value: 0	
		Format: U2.2	
	11:8	<b>LumaTDM_WT</b>	
		Default Value: 4	
		Format: U2.2	
	7:0	<b>FMD Temporal Difference Threshold</b>	
		Format: U8	
37	31:28	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	27:24	<b>Deltabit value for SHCM</b>	
		Format: U4	
		<b>Value</b>	<b>Name</b>
		5	<b>[Default]</b>
		[0,8]	Range
	23:16	<b>Coring Threshold for SHCM</b>	
		Default Value: 255	
		Format: U8	
	15:12	<b>Reserved</b>	
Access: RO			
Format: MBZ			
11:8	<b>Deltabit value for SVCM</b>		
	Format: U4		
	<b>Value</b>	<b>Name</b>	
	5	<b>[Default]</b>	
	[0,8]	Range	
7:0	<b>Coring Threshold for SVCM</b>		
	Default Value: 255		
	Format: U8		
38	31:24	<b>FMD #1 Vertical Difference Threshold</b>	
		Format: U8	
	23:16	<b>FMD #2 Vertical Difference Threshold</b>	



VEBOX_DNDI_STATE				
		Format:	U8	
39	15:14	<b>CAT Threshold</b>		
		Default Value:	0	
		Format:	U2	
	13:8	<b>FMD Tear Threshold</b>		
		Format:	U6	
	7	<b>MCDI Enable</b>		
		Use Motion Compensated Deinterlace algorithm.		
		<p style="text-align: center;"><b>Programming Notes</b></p> This bit is Ignored if DI Enable is off.		
	6:4	<b>Reserved</b>		
		Access:	RO	
Format:		MBZ		
3	<b>DN/DI Top First</b>			
	Format:	Enable		
	Indicates the top field is first in sequence, otherwise bottom is first.			
	<b>Value</b>	<b>Name</b>	<b>Description</b>	
	0		Bottom field occurs first in sequence	
1		Top field occurs first in sequence		
2:0	<b>Reserved</b>			
	Access:	RO		
	Format:	MBZ		
39	31:26	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	25	<b>FasterConvergence</b>		
		Default Value:	0	
		Format:	U1	
			For synthetic content faster convergence to current STMM value is preferred.	
	24	<b>Luma Smaller Window for TDM</b>		
		Format:	U1	
	23	<b>Chroma Smaller Window for TDM</b>		
		Format:	U1	
	22:19	<b>Neighbor Pixel Threshold</b>		
Default Value:		10		
Format:		U4		

## VEBOX\_DNDI\_STATE

	18	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	17:16	<b>Progressive Cadence Reconstruction For 2nd Field Of Previous Frame</b>		
		Format:	U2	
		<b>Value</b>	<b>Name</b>	<b>Description</b>
		0	Deinterlace	
		1	Put together with previous field in sequence	1 <sup>st</sup> field of previous frame
		2	Put together with next field in sequence	1 <sup>st</sup> field of current frame
		<b>Programming Notes</b>		
	Deflicker can be enabled only in De-interlace mode and not in Cadence construction mode.			
15:10	<b>MC Pixel Consistency Threshold</b>			
	Default Value:	25		
	Format:	U6		
9:8	<b>Progressive Cadence Reconstruction for 1st Field of Current Frame</b>			
	Format:	U2		
	<b>Value</b>	<b>Name</b>	<b>Description</b>	
	0	Deinterlace		
	1	Put together with previous field in sequence	2 <sup>nd</sup> field of previous frame	
	2	Put together with next field in sequence	2 <sup>nd</sup> field of current frame	
	<b>Programming Notes</b>			
	Deflicker can be enabled only in De-interlace mode and not if either fields are in Cadence construction mode.			
7:4	<b>SAD THB</b>			
	Default Value:	10		
	Format:	U4		
3:0	<b>SAD THA</b>			
	Default Value:	5		
	Format:	U4		
40	31:24	<b>SAD_WT[3]</b>		
		Format:	U8	
	<b>Value</b>	<b>Name</b>		
	192	Default for Natural		

<b>VEBOX_DNDI_STATE</b>								
		38 Default for Synthetic						
	23:16	<b>SAD_WT[2]</b> Format: U8 <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;">179</td> <td>Default for Natural</td> </tr> <tr> <td style="text-align: center;">25</td> <td>Default for Synthetic</td> </tr> </tbody> </table>	Value	Name	179	Default for Natural	25	Default for Synthetic
Value	Name							
179	Default for Natural							
25	Default for Synthetic							
	15:8	<b>SAD_WT[1]</b> Format: U8 <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;">166</td> <td>Default for Natural</td> </tr> <tr> <td style="text-align: center;">12</td> <td>Default for Synthetic</td> </tr> </tbody> </table>	Value	Name	166	Default for Natural	12	Default for Synthetic
Value	Name							
166	Default for Natural							
12	Default for Synthetic							
	7:0	<b>SAD_WT[0]</b> Format: U8 <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>Default for Natural and Synthetic</td> </tr> </tbody> </table>	Value	Name	0	Default for Natural and Synthetic		
Value	Name							
0	Default for Natural and Synthetic							
41	31:24	<b>Coring Threshold for Chroma SAD calculation</b> Default Value: 0 Format: U8						
	23:16	<b>Coring Threshold for Luma SAD calculation</b> Default Value: 0 Format: U8						
	15:8	<b>SAD_WT[6]</b> Format: U8 <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;">217</td> <td>Default for Natural</td> </tr> <tr> <td style="text-align: center;">64</td> <td>Default for Synthetic</td> </tr> </tbody> </table>	Value	Name	217	Default for Natural	64	Default for Synthetic
	Value	Name						
217	Default for Natural							
64	Default for Synthetic							
7:0	<b>SAD_WT[4]</b> Format: U8 <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;">218</td> <td>Default for Natural</td> </tr> <tr> <td style="text-align: center;">90</td> <td>Default for Synthetic</td> </tr> </tbody> </table>	Value	Name	218	Default for Natural	90	Default for Synthetic	
Value	Name							
218	Default for Natural							
90	Default for Synthetic							
42	31	<b>Reserved</b>						

## VEBOX\_DNDI\_STATE

		Access:	RO
		Format:	MBZ
	30	<b>Bypass Deflicker</b>	
		Format:	U1
	29	<b>PAR_UseSyntheticContentMedian</b>	
		Default Value:	0
		Format:	U1
	28	<b>PAR_LocalCheck</b>	
		Default Value:	1
		Format:	U1
	27	<b>PAR_SyntheticContentCheck</b>	
		Default Value:	0
		Format:	U1
	26:24	<b>PAR_DirectionCheckTh</b>	
		Default Value:	3
		Format:	U3
	23:16	<b>PAR_TearingLowThreshold</b>	
		Default Value:	20
		Format:	U8
	15:8	<b>PAR_TearingHighThreshold</b>	
		Default Value:	100
		Format:	U8
	7:0	<b>PAR_DiffCheckSlackThreshold</b>	
		Default Value:	15
		Format:	U8
43	31:24	<b>LPFWtLUT[3]</b>	
		Default Value:	0
		Format:	U8
	23:16	<b>LPFWtLUT[2]</b>	
		Default Value:	0
		Format:	U8
	15:8	<b>LPFWtLUT[1]</b>	
		Default Value:	0
		Format:	U8
	7:0	<b>LPFWtLUT[0]</b>	
		Default Value:	0

<b>VEBOX_DNDI_STATE</b>				
		<table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table>	Format:	U8
Format:	U8			
44	31:24	<b>LPFWtLUT[7]</b>		
		Default Value: 255		
		Format: U8		
	23:16	<b>LPFWtLUT[6]</b>		
		Default Value: 128		
		Format: U8		
	15:8	<b>LPFWtLUT[5]</b>		
		Default Value: 64		
		Format: U8		
	7:0	<b>LPFWtLUT[4]</b>		
		Default Value: 32		
		Format: U8		
45	31:20	<b>Synthetic Content Threshold</b>		
		Default Value: 1500		
		Format: U12		
	19	<b>Synthetic frame</b>		
		Default Value: 0		
		Format: enable		
	18:16	<b>TDM Harmonic Factor Natural</b>		
		Default Value: 4		
		Format: U3		
	15:13	<b>TDM Harmonic Factor Synthetic</b>		
		Default Value: 4		
		Format: U3		
12:7	<b>HV UV threshold</b>			
	Default Value: 30			
	Format: U6			
6:0	<b>TDM UV Threshold</b>			
	Default Value: 100			
	Format: U7			
46	31:29	<b>Reserved</b>		
		Access: RO		
		Format: MBZ		
	28:21	<b>Natural Content Threshold</b>		
		Default Value: 200		

## VEBOX\_DNDI\_STATE

		Format:	U8	
	20:16	<b>Max Harmonic Counter Threshold</b>		
		Default Value:	15	
		Format:	U5	
		<b>Restriction</b>		
	The maximum value that can be programmed is 15.			
	15:12	<b>Harmonic Counter Threshold</b>		
		Default Value:	8	
		Format:	U4	
	11:9	<b>SHCM Harmonic Factor Natural</b>		
		Default Value:	6	
Format:		U3		
8:6	<b>SVCM Harmonic Factor Natural</b>			
	Default Value:	6		
	Format:	U3		
5:3	<b>SHCM Harmonic Factor Synthetic</b>			
	Default Value:	4		
	Format:	U3		
2:0	<b>SVCM Harmonic Factor Synthetic</b>			
	Default Value:	4		
	Format:	U3		
47	31:0	<b>Maximum Value</b>		
		Default Value:	0x3FFFFFFF	
		Format:	U32	
48	31:20	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	19:16	<b>Num Inliner Denominator Threshold</b>		
		Default Value:	4	
		Format:	U4	
	15:12	<b>Num Inliner Numerator Threshold</b>		
		Default Value:	3	
		Format:	U4	
	11:6	<b>HV_Y Threshold</b>		
Default Value:		50		

VEBOX_DNDI_STATE			
		Format:	U6
	5:0	<b>Shifting value</b>	
		Default Value:	32
		Format:	U6



## VEBOX\_Filter\_Coefficient

VEBOX_Filter_Coefficient		
Size (in bits):	8	
Default Value:	0x00000000	
DWord	Bit	Description
0	7:0	<b>2's Complement Filter Coefficient</b> Format: S1.6 <b>Range:</b> [-2, +2)

















## VEBOX\_FORWARD\_GAMMA\_CORRECTION\_STATE

0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,  
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,

DWord	Bit	Description																																
0..2047	63:0	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <b>PRGB Corrected Value</b> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 5px;"> <tr> <td style="width: 20%;">Format:</td> <td style="text-align: center; font-weight: bold; color: red;">VEBOX_RGB_TO_GAMMA_CORRECTION</td> </tr> </table> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center; color: blue; font-weight: bold;">Programming Notes</p> <p>Order in which the values are stored:</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th colspan="10" style="color: blue; font-weight: bold;">Interleaves</th> </tr> </thead> <tbody> <tr> <td>0</td><td>256</td><td>1</td><td>257</td><td>2</td><td>258</td><td>...</td><td>...</td><td>255</td><td>511</td> </tr> <tr> <td>512</td><td>768</td><td>513</td><td>769</td><td>514</td><td>770</td><td>...</td><td>...</td><td>767</td><td>1023</td> </tr> </tbody> </table> <p>Point 0-255, 256-511 are interleaved first followed by interleaving the next set of 512 points, interleaving between points 512-767, 768-1023.</p> </div>	Format:	VEBOX_RGB_TO_GAMMA_CORRECTION	Interleaves										0	256	1	257	2	258	...	...	255	511	512	768	513	769	514	770	...	...	767	1023
Format:	VEBOX_RGB_TO_GAMMA_CORRECTION																																	
Interleaves																																		
0	256	1	257	2	258	...	...	255	511																									
512	768	513	769	514	770	...	...	767	1023																									



## 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	<b>Front End CSC Transform Enable</b>
		Format: Enable
		<b>Programming Notes</b> Single Pipe IECP Enable must also be set if this is enabled.
	30:19	<b>Reserved</b>
		Access: RO
		Format: MBZ
18:0	<b>FECS C0: Transform coefficient</b>	
	Default Value: 10000h or 1.0	
	Format: S2.16	
1	31:19	<b>Reserved</b>
		Access: RO
		Format: MBZ
	18:0	<b>FECS C1: Transform coefficient</b>
		Default Value: 0 or 0.0
		Format: S2.16
2	31:19	<b>Reserved</b>
		Access: RO
		Format: MBZ
	18:0	<b>FECS C2: Transform coefficient</b>
		Default Value: 0 or 0.0
		Format: S2.16
3	31:19	<b>Reserved</b>
		Access: RO
		Format: MBZ
	18:0	<b>FECS C3: Transform coefficient</b>
Default Value: 0 or 0.0		

<b>VEBOX_FRONT_END_CSC_STATE</b>		
		Format: S2.16
4	31:19	<b>Reserved</b>
		Access: RO
	Format: MBZ	
	18:0	<b>FECSC C4: Transform coefficient</b>
Default Value: 10000h or 1.0		
Format: S2.16		
5	31:19	<b>Reserved</b>
		Access: RO
	Format: MBZ	
	18:0	<b>FECSC C5: Transform coefficient</b>
Default Value: 0 or 0.0		
Format: S2.16		
6	31:19	<b>Reserved</b>
		Access: RO
	Format: MBZ	
	18:0	<b>FECSC C6: Transform coefficient</b>
Default Value: 0 or 0.0		
Format: S2.16		
7	31:19	<b>Reserved</b>
		Access: RO
	Format: MBZ	
	18:0	<b>FECSC C7: Transform coefficient</b>
Default Value: 0 or 0.0		
Format: S2.16		
8	31:19	<b>Reserved</b>
		Access: RO
	Format: MBZ	
	18:0	<b>FECSC C8: Transform coefficient</b>
Default Value: 10000h or 1.0		
Format: S2.16		
9	31:16	<b>FEC SC Offset out 1: Offset out for Y/R</b>
		Default Value: 0
		Format: S15
The offset value is multiplied by 2 before being added to the output.		

<b>VEBOX_FRONT_END_CSC_STATE</b>						
	15:0	<p><b>FEC SC Offset in 1: Offset in for Y/R</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					
10	31:16	<p><b>FEC SC Offset out 2: Offset out for U/G</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><b>FEC SC Offset in 2: Offset out for U/G</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					
11	31:16	<p><b>FEC SC Offset out 3: Offset out for V/B</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><b>FEC SC Offset in 3: Offset out for V/B</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\_CONTROL\_STATE

VEBOX_GAMUT_CONTROL_STATE										
Source:	VideoEnhancementCS									
Size (in bits):	576									
Default Value:	0xDA004750, 0x0000AE80, 0x00000470, 0x00000220, 0x001FFCC0, 0x0000D230, 0x00000A80, 0x001FFF40, 0x0000D6A0, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x0CD2911F, 0xB0000334, 0x00000000									
DWord	Bit	Description								
0	31:23	<b>A(r)</b>								
		<table border="1"> <tr> <td>Default Value:</td> <td>436</td> </tr> <tr> <td>Format:</td> <td>U9</td> </tr> </table> <p>Gain_factor_R (default: 436, preferred range: 256-511).</p>	Default Value:	436	Format:	U9				
	Default Value:	436								
	Format:	U9								
	22	<b>Global Mode Enable</b>								
<table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <p>The gain factor derived from state CM(w).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Advance Mode</td> </tr> <tr> <td>1</td> <td>Basic Mode</td> </tr> </tbody> </table>		Format:	U1	Value	Name	0	Advance Mode	1	Basic Mode	
Format:		U1								
Value	Name									
0	Advance Mode									
1	Basic Mode									
21	<b>Reserved</b>									
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO									
Format:	MBZ									
20:0	<b>C1</b>									
	<table border="1"> <tr> <td>Default Value:</td> <td>0004750h = 18256/65536</td> </tr> <tr> <td>Format:</td> <td>S4.16</td> </tr> </table> <p>Coefficient of 3x3 Transform matrix.</p>	Default Value:	0004750h = 18256/65536	Format:	S4.16					
	Default Value:	0004750h = 18256/65536								
Format:	S4.16									
1	31:22	<b>CM(w)</b>								
		<table border="1"> <tr> <td>Format:</td> <td>U10</td> </tr> </table> <p>WeightingFactorForGain_factor (only enabled when the GlobalModeEnable is on).</p>	Format:	U10						
	Format:	U10								
	21	<b>Reserved</b>								
		<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
Access:	RO									
Format:	MBZ									
20:0	<b>C0</b>									
	<table border="1"> <tr> <td>Default Value:</td> <td>000AE80h = 44672/65536</td> </tr> <tr> <td>Format:</td> <td>S4.16</td> </tr> </table> <p>Coefficient of 3x3 Transform matrix.</p>	Default Value:	000AE80h = 44672/65536	Format:	S4.16					
	Default Value:	000AE80h = 44672/65536								
Format:	S4.16									

<b>VEBOX_GAMUT_CONTROL_STATE</b>					
2	31:22	<b>CM(s)</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2.8</td> </tr> </table> AccurateColorComponentScaling (default: 640/256, preferred range: [512-1023]/256).	Format:	U2.8	
	Format:	U2.8			
	21	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				
20:0	<b>C3</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Default Value:</td> <td>0000470h = 1136/65536</td> </tr> <tr> <td>Format:</td> <td>S4.16</td> </tr> </table> Coefficient of 3x3 Transform matrix.	Default Value:	0000470h = 1136/65536	Format:	S4.16
Default Value:	0000470h = 1136/65536				
Format:	S4.16				
3	31:25	<b>A(g)</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U7</td> </tr> </table> Gain_factor_G (default: 26/256, preferred range: [26-127]/256).	Format:	U7	
	Format:	U7			
	24:21	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				
20:0	<b>C2</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Default Value:</td> <td>0000220h = 544/65536</td> </tr> <tr> <td>Format:</td> <td>S4.16</td> </tr> </table> Coefficient of 3x3 Transform matrix.	Default Value:	0000220h = 544/65536	Format:	S4.16
Default Value:	0000220h = 544/65536				
Format:	S4.16				
4	31:25	<b>A(b)</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U7</td> </tr> </table> Gain_factor_B (default: 26/256, preferred range: [26-127]/256).	Format:	U7	
	Format:	U7			
	24:21	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:
Access:	RO				
Format:	MBZ				
20:0	<b>C5</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 40%;">Default Value:</td> <td>1FFCC0h = -832/65536</td> </tr> <tr> <td>Format:</td> <td>S4.16</td> </tr> </table> Coefficient of 3x3 Transform matrix.	Default Value:	1FFCC0h = -832/65536	Format:	S4.16
Default Value:	1FFCC0h = -832/65536				
Format:	S4.16				
5	31:22	<b>R(s)</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U2.8</td> </tr> </table>	Format:	U2.8	
Format:	U2.8				

		<b>VEBOX_GAMUT_CONTROL_STATE</b>	
		RedScaling (default: 768/256, preferred range: [512-1023]/256).	
	21	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	20:0	<b>C4</b>	
		Default Value:	000D230h = 53808/65536
		Format:	S4.16
		Coefficient of 3x3 Transform matrix.	
6	31:24	<b>CM(i)</b>	
		Format:	U0.8
		AccurateColorComponentOffset (default: 192/256, preferred range: [0-192]/256).	
	23:21	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	20:0	<b>C7</b>	
		Default Value:	0000A80h = 2688/65536
		Format:	S4.16
		Coefficient of 3x3 Transform matrix.	
7	31:24	<b>R(i)</b>	
		Format:	U0.8
		RedOffset (default: 128/256, preferred range: [0-128]/256).	
	23:21	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	20:0	<b>C6</b>	
		Default Value:	1FFF40h = -192/65536
		Format:	S4.16
		Coefficient of 3x3 Transform matrix.	
8	31:21	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	20:0	<b>C8</b>	

<b>VEBOX_GAMUT_CONTROL_STATE</b>								
		<table border="1"> <tr> <td>Default Value:</td> <td>00D6A0h = 54944/65536</td> </tr> <tr> <td>Format:</td> <td>S4.16</td> </tr> <tr> <td colspan="2">Coefficient of 3x3 Transform matrix.</td> </tr> </table>	Default Value:	00D6A0h = 54944/65536	Format:	S4.16	Coefficient of 3x3 Transform matrix.	
Default Value:	00D6A0h = 54944/65536							
Format:	S4.16							
Coefficient of 3x3 Transform matrix.								
9	31:17	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
	Access:	RO						
Format:	MBZ							
	16:0	<b>Offset_in_R</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S16</td> </tr> <tr> <td colspan="2">The input offset for red component.</td> </tr> </table>	Default Value:	0	Format:	S16	The input offset for red component.	
Default Value:	0							
Format:	S16							
The input offset for red component.								
10	31:17	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
	Access:	RO						
Format:	MBZ							
	16:0	<b>Offset_in_G</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S16</td> </tr> <tr> <td colspan="2">The input offset for green component.</td> </tr> </table>	Default Value:	0	Format:	S16	The input offset for green component.	
Default Value:	0							
Format:	S16							
The input offset for green component.								
11	31:17	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
	Access:	RO						
Format:	MBZ							
	16:0	<b>Offset_in_B</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S16</td> </tr> <tr> <td colspan="2">The input offset for blue component.</td> </tr> </table>	Default Value:	0	Format:	S16	The input offset for blue component.	
Default Value:	0							
Format:	S16							
The input offset for blue component.								
12	31:17	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
	Access:	RO						
Format:	MBZ							
	16:0	<b>Offset_out_R</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S16</td> </tr> <tr> <td colspan="2">The output offset for red component.</td> </tr> </table>	Default Value:	0	Format:	S16	The output offset for red component.	
Default Value:	0							
Format:	S16							
The output offset for red component.								
13	31:17	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> </table>	Access:	RO				
Access:	RO							

<b>VEBOX_GAMUT_CONTROL_STATE</b>										
		<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ						
Format:	MBZ									
	16:0	<p><b>Offset_out_G</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S16</td> </tr> </table> <p>The output offset for green component.</p>	Default Value:	0	Format:	S16				
Default Value:	0									
Format:	S16									
14	31:17	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
		Access:	RO							
Format:	MBZ									
16:0	<p><b>Offset_out_B</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S16</td> </tr> </table> <p>The output offset for blue component.</p>	Default Value:	0	Format:	S16					
Default Value:	0									
Format:	S16									
15	31	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ				
		Access:	RO							
	Format:	MBZ								
	30	<p><b>FullRangeMappingEnable</b></p> <table border="1"> <tr> <td>Format:</td> <td>U1</td> </tr> </table> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Basic Mode <b>[Default]</b></td> </tr> <tr> <td style="text-align: center;">1</td> <td>Advance Mode</td> </tr> </tbody> </table>	Format:	U1	Value	Name	0	Basic Mode <b>[Default]</b>	1	Advance Mode
	Format:	U1								
	Value	Name								
	0	Basic Mode <b>[Default]</b>								
1	Advance Mode									
29:20	<p><b>d(in,default)</b></p> <table border="1"> <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	<p><b>d(out, default)</b></p> <table border="1"> <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	<p><b>d1(out)</b></p> <table border="1"> <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									



<b>VEBOX_GAMUT_CONTROL_STATE</b>																	
16	31	<b>xvYccDecEncEnable</b> Format: U1 This bit is valid only when ColorGamutCompressionnEnable is on. <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Both xvYcc decode and xvYcc encode are enabled <b>[Default]</b></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 <b>[Default]</b>	0	To disable both xvYcc decode and xvYcc encode									
		Value	Name														
		1	Both xvYcc decode and xvYcc encode are enabled <b>[Default]</b>														
		0	To disable both xvYcc decode and xvYcc encode														
	30:28	<b>CompressionLineShift</b> Format: U3 . <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>3</td> <td><b>[Default]</b></td> </tr> <tr> <td>0,4</td> <td></td> </tr> </tbody> </table>	Value	Name	3	<b>[Default]</b>	0,4										
		Value	Name														
		3	<b>[Default]</b>														
	0,4																
	27:10	<b>Reserved</b> Access: RO Format: MBZ															
		9:0 <b>d1(in)</b> Default Value: 820 Format: U10 InnerTriangleMappingLengthBelow.															
17	31:30	<b>GCC BasicModeSelection</b> Format: U2 <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Default <b>[Default]</b></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>	Value	Name	Description	00b	Default <b>[Default]</b>		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	Description													
		00b	Default <b>[Default]</b>														
		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													
	29	<b>LumaChormaOnlyCorrection</b> Format: U1 <table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Luma Only Correction <b>[Default]</b></td> </tr> <tr> <td>1</td> <td>Chorma Only Correction</td> </tr> </tbody> </table>	Value	Name	0	Luma Only Correction <b>[Default]</b>	1	Chorma Only Correction									
		Value	Name														
		0	Luma Only Correction <b>[Default]</b>														
	1	Chorma Only Correction															
28:25	<b>Reserved</b> Access: RO Format: MBZ																

<b>VEBOX_GAMUT_CONTROL_STATE</b>						
24:11	<b>BasicModeScalingFactor</b> Format: U2.12 Used when FullRangeMappingEnable is in basic mode and base mode selection bit is set to scaling factor.					
	<b>Reserved</b> Access: RO Format: MBZ					
	<b>Cpi Override</b> Format: U1 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td><b>[Default]</b></td> </tr> <tr> <td style="text-align: center;">1</td> <td>Override Cpi calculation</td> </tr> </tbody> </table>	Value	Name	0	<b>[Default]</b>	1
Value	Name					
0	<b>[Default]</b>					
1	Override Cpi calculation					

## 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	<b>Reserved</b>
		Access: RO
		Format: MBZ
	27:17	<b>Contrast</b>
		Default Value: 80h = 1.0 in fixed point U4.7
Format: U4.7 Contrast magnitude.		
16:13	<b>Reserved</b>	
	Access: RO	
	Format: MBZ	
12:1	<b>Brightness</b>	
	Default Value: 0 or 0.0	
	Format: S7.4 Brightness magnitude.	
0	<b>PROCAMP Enable</b>	
	Default Value: 1	
	Format: Enable	
1	31:16	<b>Cos_c_s</b>
		Default Value: 256
		Format: S7.8 UV multiplication cosine factor.
	15:0	<b>Sin_c_s</b>
		Default Value: 0
Format: S7.8 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>B-ch Corrected Value</b>
		Default Value: 0h
	Format: U16	
	47:32	<b>G-ch Corrected Value</b>
		Default Value: 0h
	Format: U16	
	31:16	<b>R-ch Corrected Value</b>
		Default Value: 0h
	Format: U16	
	15:0	<b>Pixel Value</b>
		Default Value: 0h
		Format: U16
		<b>Programming Notes</b>
	N indicates the index into the table. Pixel value 0 and Pixel Value 1023 should be always programmed to 0 and 0xFFFF respectively.	

## 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, 0x0007C300, 0x1291F008, 0x00094855, 0x1C1BD100, 0x03802008, 0x0002A980, 0x00080180, 0x0007CFFB, 0x18D1F07C, 0x000800BD, 0x1C080100, 0x03800000, 0x0008012B, 0x0008012B		
This state structure contains the state used by the STD/STE function.			
DWord	Bit	Description	
0	31:24	<b>V_Mid</b>	
		Default Value:	154
		Format:	U8
		Rectangle middle-point V coordinate.	
	23:16	<b>U_Mid</b>	
		Default Value:	110
		Format:	U8
		Rectangle middle-point U coordinate.	
	15:10	<b>Hue_Max</b>	
		Default Value:	14
		Format:	U6
		Rectangle half width.	
	9:4	<b>Sat_Max</b>	
		Default Value:	31
		Format:	U6
		Rectangle half length.	
	3	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	2	<b>Output Control</b>	
<b>Value</b>		<b>Name</b>	
0		Output Pixels	
1		Output STD Decisions	

VEBOX_STD_STE_STATE				
	1	<b>STE Enable</b>		
		Format:	Enable	
	0	<b>STD Enable</b>		
		Format:	Enable	
		<b>Programming Notes</b>		
This needs to be enabled if 'STD Score Output' is enabled.				
1	31	<b>STD Score Output</b>		
		Format:	Enable	
	30:28	<b>Diamond Margin</b>		
		Default Value:	4	
		Format:	U3	
	27:21	<b>Diamond_du</b>		
		Default Value:	0	
		Format:	S6	
	Rhombus center shift in the sat-direction, relative to the rectangle center.			
	20:18	<b>HS_margin</b>		
Default Value:		3		
	Format:	U3		
Defines rectangle margin.				
17:10	<b>Cos(<math>\hat{I}\pm</math>)</b>			
	Default Value:	79		
	Format:	S0.7		
The default is 79/128				
9:8	<b>Reserved</b>			
	Access:	RO		
	Format:	MBZ		
7:0	<b>Sin(<math>\hat{I}\pm</math>)</b>			
	Default Value:	101		
	Format:	S0.7		
The default is 101/128				
2	31:21	<b>Reserved</b>		
		Access:	RO	
	Format:	MBZ		

<b>VEBOX_STD_STE_STATE</b>						
	20:13	<p><b>Diamond_alpha</b></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()The default is 100/64</p>	Default Value:	100	Format:	U2.6
	Default Value:	100				
	Format:	U2.6				
12:7	<p><b>Diamond_Th</b></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><b>Diamond_dv</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> <p>Rhombus center shift in the hue-direction, relative to the rectangle center.</p>	Default Value:	0	Format:	S6	
Default Value:	0					
Format:	S6					
3	31:24	<p><b>Y_point_3</b></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><b>Y_point_2</b></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><b>Y_point_1</b></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><b>VY_STD_Enable</b></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><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
4	31:18	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> </table>	Access:	RO		
Access:	RO					

## VEBOX\_STD\_STE\_STATE

		Format:	MBZ
	17:13	<b>Y_Slope_2</b>	
		Default Value:	31
		Format:	U2.3
		Slope between points Y3 and Y4.	
		The default is 31/8	
	12:8	<b>Y_Slope_1</b>	
		Default Value:	31
		Format:	U2.3
		Slope between points Y1 and Y2.	
		The default is 31/8	
7:0	<b>Y_point_4</b>		
	Default Value:	255	
	Format:	U8	
	Fourth point of the Y piecewise linear membership function.		
5	31:16	<b>INV_Skin_types_margin</b>	
		Format:	U0.16
		1/(2* Skin_types_margin)	
		<b>Value</b>	<b>Name</b>
		20	Skin_Type_margin <b>[Default]</b>
		1638	
15:0	<b>INV_Margin_VYL</b>		
	Format:	U0.16	
		1 / Margin_VYL1/ Margin_VYL = 3300/65536	
6	31:24	<b>P1L</b>	
		Default Value:	216
		Format:	U8
		Y Point 1 of the lower part of the detection PWLF.	
	23:16	<b>P0L</b>	
		Default Value:	46
		Format:	U8
		Y Point 0 of the lower part of the detection PWLF.	



<b>VEBOX_STD_STE_STATE</b>						
	15:0	<b>INV_Margin_VYU</b> <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	<b>B1L</b> <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	<b>B0L</b> <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	<b>P3L</b> <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	<b>P2L</b> <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	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
26:16	<b>S0L</b> <table border="1"> <tr> <td>Default Value:</td> <td>7FBh</td> </tr> <tr> <td>Format:</td> <td>S2.8</td> </tr> </table> <p>Slope 0 of the lower part of the detection PWLF. The default is -5/256</p>	Default Value:	7FBh	Format:	S2.8	
Default Value:	7FBh					
Format:	S2.8					
15:8	<b>B3L</b> <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					

<b>VEBOX_STD_STE_STATE</b>						
	7:0	<b>B2L</b> <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	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: right;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
		Access:	RO			
	Format:	MBZ				
	21:11	<b>S2L</b> <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</td> </tr> </table> <p>The default is 0/256</p>	Default Value:	0	Format:	S2.8
		Default Value:	0			
	Format:	S2.8				
10:0	<b>S1L</b> <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</td> </tr> </table>	Default Value:	0	Format:	S2.8	
	Default Value:	0				
	Format:	S2.8				
<p>Slope 1 of the lower part of the detection PWLF.</p> <p>The default is 0/256</p>						
10	31:27	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: right;">RO</td> </tr> <tr> <td>Format:</td> <td style="text-align: right;">MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
		Access:	RO			
	Format:	MBZ				
	26:19	<b>P1U</b> <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	<b>P0U</b> <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	<b>S3L</b> <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</td> </tr> </table>	Default Value:	0	Format:	S2.8	
	Default Value:	0				
Format:	S2.8					
<p>Slope 3 of the lower part of the detection PWLF.</p> <p>The default is 0/256</p>						

<b>VEBOX_STD_STE_STATE</b>									
11	31:24	<b>B1U</b> <table border="1"> <tr> <td>Default Value:</td> <td>163</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 1 of the upper part of the detection PWLF.</p>	Default Value:	163	Format:	U8			
		Default Value:	163						
		Format:	U8						
		23:16	<b>B0U</b> <table border="1"> <tr> <td>Default Value:</td> <td>143</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 0 of the upper part of the detection PWLF.</p>	Default Value:	143	Format:	U8		
Default Value:	143								
Format:	U8								
15:8	<b>P3U</b> <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 upper part of the detection PWLF.</p>	Default Value:	236	Format:	U8				
	Default Value:	236							
Format:	U8								
7:0	<b>P2U</b> <table border="1"> <tr> <td>Default Value:</td> <td>150</td> </tr> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>Y Point 2 of the upper part of the detection PWLF.</p>	Default Value:	150	Format:	U8				
Default Value:	150								
Format:	U8								
12	31:27	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ			
		Access:	RO						
	Format:	MBZ							
	26:16	<b>S0U</b> <table border="1"> <tr> <td>Default Value:</td> <td>256</td> </tr> <tr> <td>Format:</td> <td>S2.8</td> </tr> </table> <p>Slope 0 of the upper part of the detection PWLF. The default is 256/256</p>	Default Value:	256	Format:	S2.8			
Default Value:		256							
Format:	S2.8								
15:8	<b>B3U</b> <table border="1"> <tr> <td>Format:</td> <td>U8</td> </tr> </table> <p>V Bias 3 of the upper part of the detection PWLF.</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>200</td> <td>[Default]</td> </tr> <tr> <td>140</td> <td></td> </tr> </tbody> </table>	Format:	U8	Value	Name	200	[Default]	140	
	Format:	U8							
	Value	Name							
200	[Default]								
140									
7:0	<b>B2U</b> <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								

<b>VEBOX_STD_STE_STATE</b>			
13	31:22	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	21:11	<b>S2U</b>	
		Default Value: 74Dh	
		Format: S2.8	
		Slope 2 of the upper part of the detection PWLF. The default is -179/256	
	10:0	<b>S1U</b>	
		Default Value: 113	
Format: S2.8			
Slope 1 of the upper part of the detection PWLF. The default is 113/256			
14	31:28	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	27:20	<b>Skin_types_margin</b>	
		Default Value: 20	
		Format: U8	
		Skin types Y marginRestrict Skin_types_thresh >= Skin_types_margin > 0Restrict (Skin_types_thresh + Skin_types_margin) <= 255	
	19:12	<b>Skin_types_thresh</b>	
		Default Value: 120	
		Format: U8	
		Skin types Y marginRestrict Skin_types_thresh >= Skin_types_margin > 0Restrict (Skin_types_thresh + Skin_types_margin) <= 255	
	11	<b>Skin_Types_Enable</b>	
		Default Value: 0 Disable	
Format: Enable Treat differently bright and dark skin types			
10:0	<b>S3U</b>		
	Default Value: 0		

<b>VEBOX_STD_STE_STATE</b>												
		<table border="1"> <tr> <td>Format:</td> <td>S2.8</td> </tr> <tr> <td colspan="2">Slope 3 of the upper part of the detection PWLF.</td> </tr> <tr> <td colspan="2">The default is 0/256</td> </tr> </table>	Format:	S2.8	Slope 3 of the upper part of the detection PWLF.		The default is 0/256					
Format:	S2.8											
Slope 3 of the upper part of the detection PWLF.												
The default is 0/256												
15	31	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
	Access:	RO										
	Format:	MBZ										
	30:21	<p><b>SATB1</b></p> <table border="1"> <tr> <td>Format:</td> <td>S7.2</td> </tr> <tr> <td colspan="2">First bias for the saturation PWLF (bright skin).</td> </tr> <tr> <td colspan="2">The default numerical value is -8/4</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>3F8h</td> <td></td> </tr> </table>	Format:	S7.2	First bias for the saturation PWLF (bright skin).		The default numerical value is -8/4		Value	Name	3F8h	
	Format:	S7.2										
	First bias for the saturation PWLF (bright skin).											
The default numerical value is -8/4												
Value	Name											
3F8h												
20:14	<p><b>SATP3</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>31</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> <tr> <td colspan="2">Third point for the saturation PWLF (bright skin).</td> </tr> </table>	Default Value:	31	Format:	S6	Third point for the saturation PWLF (bright skin).						
Default Value:	31											
Format:	S6											
Third point for the saturation PWLF (bright skin).												
13:7	<p><b>SATP2</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>6</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> <tr> <td colspan="2">Second point for the saturation PWLF (bright skin).</td> </tr> </table>	Default Value:	6	Format:	S6	Second point for the saturation PWLF (bright skin).						
Default Value:	6											
Format:	S6											
Second point for the saturation PWLF (bright skin).												
6:0	<p><b>SATP1</b></p> <table border="1"> <tr> <td>Format:</td> <td>S6</td> </tr> <tr> <td colspan="2">First point for the saturation PWLF (bright skin).</td> </tr> <tr> <td colspan="2">The default numerical value is -6/64.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td>7Ah</td> <td></td> </tr> </table>	Format:	S6	First point for the saturation PWLF (bright skin).		The default numerical value is -6/64.		Value	Name	7Ah		
Format:	S6											
First point for the saturation PWLF (bright skin).												
The default numerical value is -6/64.												
Value	Name											
7Ah												
16	31	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ						
	Access:	RO										
	Format:	MBZ										
30:20	<p><b>SATS0</b></p> <table border="1"> <tr> <td>Default Value:</td> <td>297</td> </tr> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table>	Default Value:	297	Format:	U3.8							
Default Value:	297											
Format:	U3.8											

## VEBOX\_STD\_STE\_STATE

		Zeroth slope for the saturation PWLF (bright skin)	
		The default is 297/256	
	19:10	<b>SATB3</b>	
		Default Value:	124
		Format:	S7.2
		Third bias for the saturation PWLF (bright skin)	
		The default is 124/4	
	9:0	<b>SATB2</b>	
		Default Value:	8
		Format:	S7.2
		Second bias for the saturation PWLF (bright skin)	
		The default is 8/4	
17	31:22	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	21:11	<b>SATS2</b>	
		Default Value:	297
		Format:	U3.8
		Second slope for the saturation PWLF (bright skin)	
		The default is 297/256	
	10:0	<b>SATS1</b>	
		Default Value:	85
		Format:	U3.8
		First slope for the saturation PWLF (bright skin)	
		The default is 85/256	
18	31:25	<b>HUEP3</b>	
		Default Value:	14
		Format:	S6
		Third point for the hue PWLF (bright skin)	
	24:18	<b>HUEP2</b>	
		Default Value:	6
		Format:	S6
		Second point for the hue PWLF (bright skin)	

VEBOX_STD_STE_STATE								
	17:11	<b>HUEP1</b> Default Value: 7Ah -6 Format: S6 First point for the hue PWLF (bright skin)						
		<b>SATS3</b> Default Value: 256 Format: U3.8 Third slope for the saturation PWLF (bright skin) The default is 256/256						
19	31:30	<b>Reserved</b> Access: RO Format: MBZ						
	29:20	<b>HUEB3</b> Default Value: 56 Format: S7.2 Third bias for the hue PWLF (bright skin) The default is 56/4						
	19:10	<b>HUEB2</b> Default Value: 8 Format: S7.2 Second bias for the hue PWLF (bright skin) The default is 8/4						
	9:0	<b>HUEB1</b> Format: S7.2 First bias for the hue PWLF (bright skin) The default is 8/4						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>[Default]</td> </tr> <tr> <td>0xf8</td> <td></td> </tr> </tbody> </table>	Value	Name	8	[Default]	0xf8	
		Value	Name					
8	[Default]							
0xf8								
<b>Reserved</b> Access: RO								
20	31:22	<b>Reserved</b> Access: RO						

VEBOX_STD_STE_STATE			
		Format:	MBZ
	21:11	<b>HUES1</b>	
		Default Value:	85
		Format:	U3.8
		First slope for the hue PWLF (bright skin)	
		The default is 85/256	
	10:0	<b>HUES0</b>	
		Default Value:	384
		Format:	U3.8
		Zeroth slope for the hue PWLF (bright skin)	
		The default is 384/256	
21	31:22	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	21:11	<b>HUES3</b>	
		Default Value:	256
		Format:	U3.8
		Third slope for the hue PWLF (bright skin)	
		The default is 256/256	
	10:0	<b>HUES2</b>	
		Default Value:	384
		Format:	U3.8
		Second slope for the hue PWLF (bright skin)	
		The default is 384/256	
22	31	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	30:21	<b>SATB1_DARK</b>	
		Default Value:	0
		Format:	S7.2
		First bias for the saturation PWLF (dark skin)	
		The default is 0/4	



<b>VEBOX_STD_STE_STATE</b>			
	20:14	<b>SATP3_DARK</b>	
		Default Value: 31	
		Format: S6	
			Third point for the saturation PWLF (dark skin)
	13:7	<b>SATP2_DARK</b>	
		Default Value: 31	
		Format: S6	
			Second point for the saturation PWLF (dark skin)
	6:0	<b>SATP1_DARK</b>	
Default Value: 7Bh			
Format: S6			
		First point for the saturation PWLF (dark skin) Default Value: -5	
23	31	<b>Reserved</b>	
		Access: RO	
		Format: MBZ	
	30:20	<b>SATS0_DARK</b>	
		Default Value: 397	
		Format: U3.8	
		Zeroth slope for the saturation PWLF (dark skin)	
		The default is 397/256	
	19:10	<b>SATB3_DARK</b>	
		Default Value: 124	
		Format: S7.2	
		Third bias for the saturation PWLF (dark skin)	
The default is 124/4			
9:0	<b>SATB2_DARK</b>		
	Default Value: 124		
	Format: S7.2		
	Second bias for the saturation PWLF (dark skin)		
	The default is 124/4		
24	31:22	<b>Reserved</b>	
		Access: RO	

<b>VEBOX_STD_STE_STATE</b>						
		<table border="1"> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Format:	MBZ		
Format:	MBZ					
	21:11	<b>SATS2_DARK</b> <table border="1"> <tr> <td>Default Value:</td> <td>256</td> </tr> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table> <p>Second slope for the saturation PWLF (dark skin) The default is 256/256</p>	Default Value:	256	Format:	U3.8
Default Value:	256					
Format:	U3.8					
	10:0	<b>SATS1_DARK</b> <table border="1"> <tr> <td>Default Value:</td> <td>189</td> </tr> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table> <p>First slope for the saturation PWLF (dark skin) The default is 189/256</p>	Default Value:	189	Format:	U3.8
Default Value:	189					
Format:	U3.8					
25	31:25	<b>HUEP3_DARK</b> <table border="1"> <tr> <td>Default Value:</td> <td>14</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> <p>Third point for the hue PWLF (dark skin).</p>	Default Value:	14	Format:	S6
	Default Value:	14				
	Format:	S6				
	24:18	<b>HUEP2_DARK</b> <table border="1"> <tr> <td>Default Value:</td> <td>2</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> <p>Second point for the hue PWLF (dark skin).</p>	Default Value:	2	Format:	S6
Default Value:	2					
Format:	S6					
17:11	<b>HUEP1_DARK</b> <table border="1"> <tr> <td>Default Value:</td> <td>0</td> </tr> <tr> <td>Format:</td> <td>S6</td> </tr> </table> <p>First point for the hue PWLF (dark skin).</p>	Default Value:	0	Format:	S6	
Default Value:	0					
Format:	S6					
10:0	<b>SATS3_DARK</b> <table border="1"> <tr> <td>Default Value:</td> <td>256</td> </tr> <tr> <td>Format:</td> <td>U3.8</td> </tr> </table> <p>Third slope for the saturation PWLF (dark skin) The default is 256/256</p>	Default Value:	256	Format:	U3.8	
Default Value:	256					
Format:	U3.8					
26	31:30	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
29:20	<b>HUEB3_DARK</b>					

VEBOX_STD_STE_STATE				
		Default Value:	56	
		Format:	S7.2	
		Third bias for the hue PWLF (dark skin).		
		The default is 56/4		
	19:10	<b>HUEB2_DARK</b>		
		Default Value:	0	
		Format:	S7.2	
		Second bias for the hue PWLF (dark skin).		
	The default is 0/4			
	9:0	<b>HUEB1_DARK</b>		
		Default Value:	0	
		Format:	S7.2	
First bias for the hue PWLF (dark skin).				
The default is 0/4				
27	31:22	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	21:11	<b>HUES1_DARK</b>		
		Default Value:	256	
		Format:	U3.8	
		First slope for the hue PWLF (dark skin).		
	The default is 256/256			
	10:0	<b>HUES0_DARK</b>		
		Format:	U3.8	
Zeroth slope for the hue PWLF (dark skin).				
The default is 299/256				
<b>Value</b>		<b>Name</b>		
299		[Default]		
256				
28	31:22	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	

<b>VEBOX_STD_STE_STATE</b>		
	21:11	<b>HUES3_DARK</b>
		Default Value: 256
		Format: U3.8
		Third slope for the hue PWLF (dark skin).
		The default is 256/256
	10:0	<b>HUES2_DARK</b>
		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	<b>SatFactor3</b>						
		Format: U1.7						
		The saturation factor for yellow.						
		The default is 220/128						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>220</td> <td>[Default]</td> </tr> <tr> <td>160</td> <td></td> </tr> </tbody> </table>	Value	Name	220	[Default]	160	
		Value	Name					
	220	[Default]						
	160							
	23:16	<b>SatFactor2</b>						
		Format: U1.7						
		The saturation factor for red.						
		The default is 220/128						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>220</td> <td>[Default]</td> </tr> <tr> <td>160</td> <td></td> </tr> </tbody> </table>	Value	Name	220	[Default]	160	
		Value	Name					
220	[Default]							
160								
15:8	<b>SatFactor1</b>							
	Format: U1.7							
	The saturation factor for magenta.							
	The default is 220/128							
	<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>220</td> <td>[Default]</td> </tr> <tr> <td>160</td> <td></td> </tr> </tbody> </table>	Value	Name	220	[Default]	160		
	Value	Name						
220	[Default]							
160								
7	<b>TCC Enable</b>							
	Format: Enable							
6:0	<b>Reserved</b>							

VEBOX_TCC_STATE			
		Access:	RO
		Format:	MBZ
1	31:24	<b>SatFactor6</b>	
		Format:	U1.7
		The saturation factor for blue.	
		The default is 220/128	
		<b>Value</b>	<b>Name</b>
		220	[Default]
	160		
	23:16	<b>SatFactor5</b>	
		Format:	U1.7
		The saturation factor for cyan.	
		The default is 220/128	
		<b>Value</b>	<b>Name</b>
		220	[Default]
	160		
	15:8	<b>SatFactor4</b>	
		Format:	U1.7
The saturation factor for green.			
The default is 220/128			
<b>Value</b>		<b>Name</b>	
220		[Default]	
160			
7:0	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
2	31:30	<b>Reserved</b>	
		Access:	RO
	Format:	MBZ	
	29:20	<b>BaseColor3</b>	
Default Value:		483	
	Format:	U10	

<b>VEBOX_TCC_STATE</b>						
		Base Color 3 - this value must be greater than BaseColor2				
	19:10	<b>BaseColor2</b> <table border="1"> <tr> <td>Default Value:</td> <td>307</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> Base Color 2 - this value must be greater than BaseColor1	Default Value:	307	Format:	U10
Default Value:	307					
Format:	U10					
	9:0	<b>BaseColor1</b> <table border="1"> <tr> <td>Default Value:</td> <td>145</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> Base Color 1	Default Value:	145	Format:	U10
Default Value:	145					
Format:	U10					
3	31:30	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	29:20	<b>BaseColor6</b> <table border="1"> <tr> <td>Default Value:</td> <td>995</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> Base Color 6 - this value must be greater than BaseColor5	Default Value:	995	Format:	U10
Default Value:	995					
Format:	U10					
19:10	<b>BaseColor5</b> <table border="1"> <tr> <td>Default Value:</td> <td>819</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> Base Color 5 - this value must be greater than BaseColor4	Default Value:	819	Format:	U10	
Default Value:	819					
Format:	U10					
9:0	<b>BaseColor4</b> <table border="1"> <tr> <td>Default Value:</td> <td>657</td> </tr> <tr> <td>Format:</td> <td>U10</td> </tr> </table> Base Color 4 - this value must be greater than BaseColor3	Default Value:	657	Format:	U10	
Default Value:	657					
Format:	U10					
4	31:16	<b>ColorTransitSlope23</b> <table border="1"> <tr> <td>Default Value:</td> <td>744</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> The calculation result of $1 / (BC3 - BC2)$ [1/62]	Default Value:	744	Format:	U0.16
	Default Value:	744				
Format:	U0.16					
15:0	<b>ColorTransitSlope2</b> <table border="1"> <tr> <td>Default Value:</td> <td>405</td> </tr> <tr> <td>Format:</td> <td>U0.16</td> </tr> </table> The calculation result of $1 / (BC2 - BC1)$ [1/57]	Default Value:	405	Format:	U0.16	
Default Value:	405					
Format:	U0.16					

<b>VEBOX_TCC_STATE</b>						
5	31:16	<b>ColorTransitSlope45</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">407</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U0.16</td> </tr> </table> <p>The calculation result of <math>1 / (BC5 - BC4)</math> [1/57]</p>	Default Value:	407	Format:	U0.16
		Default Value:	407			
Format:	U0.16					
15:0	<b>ColorTransitSlope34</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">1131</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U0.16</td> </tr> </table> <p>The calculation result of <math>1 / (BC4 - BC3)</math> [1/61]</p>	Default Value:	1131	Format:	U0.16	
		Default Value:	1131			
Format:	U0.16					
6	31:16	<b>ColorTransitSlope61</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">377</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U0.16</td> </tr> </table> <p>The calculation result of <math>1 / (BC1 - BC6)</math> [1/62]</p>	Default Value:	377	Format:	U0.16
		Default Value:	377			
Format:	U0.16					
15:0	<b>ColorTransitSlope56</b> <table border="1" style="width: 100%;"> <tr> <td>Default Value:</td> <td style="text-align: center;">372</td> </tr> <tr> <td>Format:</td> <td style="text-align: center;">U0.16</td> </tr> </table> <p>The calculation result of <math>1 / (BC6 - BC5)</math> [1/62]</p>	Default Value:	372	Format:	U0.16	
		Default Value:	372			
Format:	U0.16					
7	31:22	<b>ColorBias3</b> <table border="1" style="width: 100%;"> <tr> <td>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	<b>ColorBias2</b> <table border="1" style="width: 100%;"> <tr> <td>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	<b>ColorBias1</b> <table border="1" style="width: 100%;"> <tr> <td>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	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td style="text-align: center;">RO</td> </tr> </table>	Access:	RO			
Access:	RO					



VEBOX_TCC_STATE			
		Format:	MBZ
8	31:22	<b>ColorBias6</b>	
		Default Value:	0
		Format:	U2.8
		Color bias for BaseColor6.	
	21:12	<b>ColorBias5</b>	
		Default Value:	0
		Format:	U2.8
		Color bias for BaseColor5.	
	11:2	<b>ColorBias4</b>	
		Default Value:	0
		Format:	U2.8
		Color bias for BaseColor4.	
1:0	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
9	31	<b>Reserved</b>	
		Access:	RO
	30:24	<b>UV Threshold</b>	
		Default Value:	3
		Format:	U7
		Low UV threshold.	
	23:19	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	18:16	<b>UV Threshold Bits</b>	
		Default Value:	3
		Format:	U3
Low UV transition width bits.			
15:13	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	

<b>VEBOX_TCC_STATE</b>		
	12:8	<b>STE Threshold</b>
		Default Value: 0
		Format: U5
	Skin tone pixels enhancement threshold.	
	7:3	<b>Reserved</b>
		Access: RO
Format: MBZ		
2:0	<b>STE Slope Bits</b>	
	Default Value: 0	
	Format: U3	
Skin tone pixels enhancement slope bits.		
10	31:16	<b>Inv_UVMaxColor</b>
		Default Value: 146
		Format: U16
	1 / UVMaxColor. Used for the SFs2 calculation.	
	15:9	<b>Reserved</b>
		Access: RO
		Format: MBZ
	8:0	<b>UVMaxColor</b>
		Default Value: 448
Format: U9		
The maximum absolute value of the legal UV pixels. Used for the SFs2 calculation.		

## VEBOX\_VERTEX\_TABLE\_ENTRY

VEBOX_VERTEX_TABLE_ENTRY				
Source:	VideoEnhancementCS			
Size (in bits):	32			
Default Value:	0x00000000			
DWord	Bit	Description		
0	31:28	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	27:16	<b>Vertex table entry 0 - Lv (12 bits)</b>		
		<b>Value</b>	<b>Name</b>	<b>Description</b>
		100h-ED6h		Range for Vertices BT601 and BT709
	15:12	<b>Reserved</b>		
		Access:	RO	
		Format:	MBZ	
	11:0	<b>Vertex table entry 0 - Cv (12 bits)</b>		
<b>Value</b>		<b>Name</b>	<b>Description</b>	
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	15:12	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	11	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	10:3	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	2	<b>Command Privilege Violation Error</b> 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	<b>Reserved</b>		
	Access:	RO	
	Format:	MBZ	
0	<b>Instruction Error</b> 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"> <li>Client ID value (Bits 31:29 of the Header) is not supported (only MI, 2D and 3D are supported).</li> <li>Defeatured MI Instruction Opcodes:</li> </ul>		
	<b>Value</b>	<b>Name</b>	<b>Description</b>
	1		Instruction Error detected
	<b>Programming Notes</b>		
	This error indications cannot be cleared except by reset (i.e., it is a fatal error).		

## VERTEX\_BUFFER\_STATE

<b>VERTEX_BUFFER_STATE</b>				
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	<b>Vertex Buffer Index</b>		
		Format: U6		
		This field contains an index value which selects the VB state being defined.		
		<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>[0,32]</td> <td></td> </tr> </tbody> </table>	Value	Name
Value	Name			
[0,32]				
25		<b>L3 Bypass Disable</b>		
		Format: Disable		
		<p style="text-align: center;"><b>Description</b></p> <p>When set, vertex data fetches from this vertex buffer are routed through the L3 and therefore that vertex data may be coherent with the L3 cache, as controlled by the Memory Object Control State (MOCS) value. I.e., if portions of the vertex buffer already reside in the L3 (e.g., were written or read by another L3 agent), reads from VF may hit in the L3 with the cached data returned to VF. If reads from VF miss in the L3 cache, the reads are directed to the next higher in the memory hierarchy, but the data returned in not placed in the L3 cache. The MOCS value must not be set to cache the data in L3.</p> <p>When clear, vertex data reads from this vertex buffer bypass the L3 logic, therefore precluding the coherency of that data in the L3 cache. If the vertex buffer data can becached in L3, the L3 cache must first be flushed to maintain vertex buffer data coherency.</p>		
24:23		<b>Reserved</b>		
		<table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
22:16		<b>Memory Object Control State</b>		
		<table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td><b>MEMORY_OBJECT_CONTROL_STATE</b></td> </tr> </table> <p>Specifies the memory object control state for this vertex buffer.</p>	Format:	<b>MEMORY_OBJECT_CONTROL_STATE</b>
Format:	<b>MEMORY_OBJECT_CONTROL_STATE</b>			
15		<b>Reserved</b>		
		<table border="1" style="width: 100%;"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO
Access:	RO			
Format:	MBZ			
14		<b>Address Modify Enable</b>		

<b>VERTEX_BUFFER_STATE</b>										
	<p>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.</p>									
13	<p><b>Null Vertex Buffer</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>Enable</td> </tr> </table> <p>This field enabled causes any fetch for vertex data to return 0.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <p>VERTEX_BUFFER_STATE.Null Vertex Buffer must be set when the VERTEX_BUFFER_STATE.Buffer Size is 0x0.</p>	Format:	Enable	<b>Programming Notes</b>						
Format:	Enable									
<b>Programming Notes</b>										
12	<p><b>Reserved</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ					
Access:	RO									
Format:	MBZ									
11:0	<p><b>Buffer Pitch</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>U12</td> </tr> </table> <p>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.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <thead> <tr> <th style="width: 30%;">Value</th> <th style="width: 30%;">Name</th> <th style="width: 40%;">Description</th> </tr> </thead> <tbody> <tr> <td>[0,4095]</td> <td></td> <td>Bytes</td> </tr> </tbody> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <ul style="list-style-type: none"> <li>Different VERTEX_BUFFER_STATE structures can refer to the same memory region using different Buffer Pitch values.</li> <li>See note on 64-bit float alignment in Buffer Starting Address.</li> </ul>	Format:	U12	Value	Name	Description	[0,4095]		Bytes	<b>Programming Notes</b>
Format:	U12									
Value	Name	Description								
[0,4095]		Bytes								
<b>Programming Notes</b>										
1..2	<p>63:0 <b>Buffer Starting Address</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Format:</td> <td>GraphicsAddress[63:0]</td> </tr> </table> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Description</b></td> </tr> </table> <p>This field contains the byte-aligned Graphics Address 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> <p>GraphicsAddress [63:48] are ignored by the HW.</p> <table border="1" style="width: 100%; background-color: #e6f2ff;"> <tr> <td style="text-align: center;"><b>Programming Notes</b></td> </tr> </table> <ul style="list-style-type: none"> <li>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.</li> <li>VBs can only be allocated in linear (not tiled) graphics memory.</li> </ul>	Format:	GraphicsAddress[63:0]	<b>Description</b>	<b>Programming Notes</b>					
Format:	GraphicsAddress[63:0]									
<b>Description</b>										
<b>Programming Notes</b>										

<b>VERTEX_BUFFER_STATE</b>								
		<ul style="list-style-type: none"> <li>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).</li> </ul>						
3	31:0	<p><b>Buffer Size</b></p> <table border="1" style="width: 100%;"> <tr> <td>Format:</td> <td>U32</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%; text-align: center;"> <thead> <tr> <th style="width: 70%;">Value</th> <th style="width: 30%;">Name</th> </tr> </thead> <tbody> <tr> <td>[0, FFFFFFFFh]</td> <td></td> </tr> </tbody> </table>	Format:	U32	Value	Name	[0, FFFFFFFFh]	
Format:	U32							
Value	Name							
[0, FFFFFFFFh]								

## VERTEX\_ELEMENT\_STATE

<b>VERTEX_ELEMENT_STATE</b>										
Source:	RenderCS									
Size (in bits):	64									
Default Value:	0x00000000, 0x00000000									
<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>										
<b>Programming Notes</b>										
<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>When SourceElementFormat is set to one of the *64*_PASSTHRU formats then VFCOMP_STORE_SRC must be used for every valid component.</li> <li>Any SourceElementFormat of *64*_PASSTHRU cannot be used with an element which has edge flag enabled.</li> </ul>										
<p>The SourceElementFormat needs to be a single-component format with an element which has edge flag enabled.</p>										
<p>Software shall not attempt to disable any components (via 3DSTATE_VF_COMPONENT_PACKING) for elements associated with 256-bit SURFACE_FORMATs.</p>										
DWord	Bit	Description								
0	31:26	<p><b>Vertex Buffer Index</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 70%;">Format:</td> <td style="text-align: center;">U6</td> </tr> <tr> <td colspan="2">This field specifies which vertex buffer the element is sourced from.</td> </tr> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> <tr> <td style="text-align: center;">[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									



<b>VERTEX_ELEMENT_STATE</b>												
	<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> </thead> <tbody> <tr> <td colspan="2">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.</td> </tr> </tbody> </table>	<b>Programming Notes</b>		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.								
<b>Programming Notes</b>												
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	<p><b>Valid</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Boolean</td> </tr> </table> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">Value</th> <th style="width: 15%;">Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1h</td> <td>TRUE</td> <td>this vertex element is used in vertex assembly</td> </tr> <tr> <td>0h</td> <td>FALSE</td> <td>this vertex element is not used.</td> </tr> </tbody> </table>	Format:	Boolean	Value	Name	Description	1h	TRUE	this vertex element is used in vertex assembly	0h	FALSE	this vertex element is not used.
Format:	Boolean											
Value	Name	Description										
1h	TRUE	this vertex element is used in vertex assembly										
0h	FALSE	this vertex element is not used.										
24:16	<p><b>Source Element Format</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td><b>SURFACE_FORMAT</b></td> </tr> </table> <p>Range: Valid formats are found in the 3D Primitive Processing FormatConversion portion of the vertex fetch chapter.</p> <p>Format:</p> <p>The encoding of this field is identical the Surface Format field of the SURFACE_STATE structure, as described in the Sampler chapter.</p> <p>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).</p>	Format:	<b>SURFACE_FORMAT</b>									
Format:	<b>SURFACE_FORMAT</b>											
15	<p><b>Edge Flag Enable</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Format:</td> <td>Enable</td> </tr> </table> <p>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.</p> <ul style="list-style-type: none"> <li>• 3DPRIM_TRILIST*</li> <li>• 3DPRIM_POLYGON</li> <li>• 3DPRIM_QUADLIST</li> </ul> <p>If this bit is DISABLED for all valid VERTEX_ELEMENTS, the vertex will be assigned a default EdgeFlag of TRUE.</p> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th colspan="2" style="text-align: center;"><b>Programming Notes</b></th> </tr> </thead> <tbody> <tr> <td colspan="2"> <ul style="list-style-type: none"> <li>• This bit must only be ENABLED on the last valid VERTEX_ELEMENT structure.</li> <li>• When set, Component 0 Control must be set to VFCOMP_STORE_SRC, and Component 1-3 Control must be set to VFCOMP_NOSTORE.</li> </ul> </td> </tr> </tbody> </table>	Format:	Enable	<b>Programming Notes</b>		<ul style="list-style-type: none"> <li>• This bit must only be ENABLED on the last valid VERTEX_ELEMENT structure.</li> <li>• When set, Component 0 Control must be set to VFCOMP_STORE_SRC, and Component 1-3 Control must be set to VFCOMP_NOSTORE.</li> </ul>						
Format:	Enable											
<b>Programming Notes</b>												
<ul style="list-style-type: none"> <li>• This bit must only be ENABLED on the last valid VERTEX_ELEMENT structure.</li> <li>• When set, Component 0 Control must be set to VFCOMP_STORE_SRC, and Component 1-3 Control must be set to VFCOMP_NOSTORE.</li> </ul>												
14:12	<b>Reserved</b>											

<b>VERTEX_ELEMENT_STATE</b>							
	<table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
Access:	RO						
Format:	MBZ						
11:0	<p><b>Source Element Offset</b></p> <table border="1"> <tr> <td>Format:</td> <td>U12</td> </tr> </table> <p>Byte offset of the source vertex element data in the structures comprising the vertex buffer.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Value</th> <th style="text-align: center;">Name</th> </tr> </thead> <tbody> <tr> <td>[0,2047]</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"><b>Programming Notes</b></p> <p>See note on 64-bit float alignment in Buffer Starting Address.</p>	Format:	U12	Value	Name	[0,2047]	
Format:	U12						
Value	Name						
[0,2047]							
1	31	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO					
	Format:	MBZ					
	30:28	<p><b>Component 0 Control</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Vertex_Component_Control</b></td> </tr> </table> <p>Refer to the 3D_Vertex_Component_Control table below</p>	Format:	<b>3D_Vertex_Component_Control</b>			
	Format:	<b>3D_Vertex_Component_Control</b>					
	27	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
	Access:	RO					
	Format:	MBZ					
	26:24	<p><b>Component 1 Control</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Vertex_Component_Control</b></td> </tr> </table> <p>Refer to the 3D_Vertex_Component_Control table below</p>	Format:	<b>3D_Vertex_Component_Control</b>			
	Format:	<b>3D_Vertex_Component_Control</b>					
23	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
Access:	RO						
Format:	MBZ						
22:20	<p><b>Component 2 Control</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Vertex_Component_Control</b></td> </tr> </table> <p>Refer to the 3D_Vertex_Component_Control table below</p>	Format:	<b>3D_Vertex_Component_Control</b>				
Format:	<b>3D_Vertex_Component_Control</b>						
19	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ		
Access:	RO						
Format:	MBZ						
18:16	<p><b>Component 3 Control</b></p> <table border="1"> <tr> <td>Format:</td> <td><b>3D_Vertex_Component_Control</b></td> </tr> </table> <p>Refer to the 3D_Vertex_Component_Control table below</p>	Format:	<b>3D_Vertex_Component_Control</b>				
Format:	<b>3D_Vertex_Component_Control</b>						

VERTEX_ELEMENT_STATE		
	15:0	<b>Reserved</b>
		Access: RO
		Format: MBZ



## Vertical Line Stride Override Message Descriptor Control Field

<b>MDC_VLSO - Vertical Line Stride Override Message Descriptor Control Field</b>				
Size (in bits):		3		
Default Value:		0x00000000		
DWord	Bit	Description		
0	2	<p><b>Vertical Line Stride Override</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>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><b>Vertical Line Stride</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>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><b>Vertical Line Stride Offset</b></p> <table border="1" style="width: 100%;"> <tr> <td style="width: 60%;">Format:</td> <td>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			

## VideoDecoder Interrupt Vector

VIDEODECODER_INTR_VEC - VideoDecoder Interrupt Vector						
Size (in bits):	16					
Default Value:	0x00000000					
DWord	Bit	Description				
0	15	<p><b>Catastrophic Error</b></p> <p>This interrupt signals that a unrecoverable error (for e.g encountered fault when accessing a page mapped in Global GTT) during the engine processing. When Memory interface signals this error, the Command Streamer will stop parsing any more instructions. Scheduler is expected to reset the engine to evict the context</p>				
	14:12	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	11	<p><b>VCS Wait On Semaphore</b></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. Ring Buffer Scheduling: Set when MI_SEMAPHORE_WAIT command is un-successful.</p>				
	10	<p><b>Reserved</b></p> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	9	<p><b>CS TR Invalid Tile Detection</b></p>				
	8	<p><b>VCS Context Switch Interrupt</b></p> <p>Set when a context switch has just occurred. <b>Execlist Enable bit</b> needs to be set for this interrupt to occur.</p>				
7	<p><b>Legacy Context Per Process Page Fault Interrupt</b></p> <p>Fault interrupt is generated by GA fabric, not by the CS This interrupt is for handling Legacy context PP GTT Page Fault.</p>					
6	<p><b>VCS Watchdog Counter Expired</b></p> <p>Set when the VCS timeout counter has reached the timeout thresh-hold value.</p>					
5	<p><b>Reserved</b></p>					
4	<p><b>VCS MI Flush DW Notify</b></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><b>VCS Error Interrupt</b></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</p>					

## VIDEODECODER\_INTR\_VEC - VideoDecoder Interrupt Vector

		<p>the "Error Mask Register" determine which error conditions will cause the error status bit to be set and the interrupt to occur.</p> <p><b>Page Table Error:</b> Indicates a page table error.  <b>Instruction Parser Error:</b> The Blitter Instruction Parser encounters an error while parsing an instruction.</p>	
	2:1	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
0		<p><b>VCS1 MI User Interrupt</b></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>	

## VideoEnhancement Interrupt Vector

VIDEOENHANCE_INTR_VEC - VideoEnhancement Interrupt Vector						
Size (in bits):	16					
Default Value:	0x00000000					
DWord	Bit	Description				
0	15	<b>Catastrophic Error</b> This interrupt signals that a unrecoverable error (for e.g encountered fault when accessing a page mapped in Global GTT) during the engine processing. When Memory interface signals this error, the Command Streamer will stop parsing any more instructions. Scheduler is expected to reset the engine to evict the context				
	14:12	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	11	<b>VECS Wait On Semaphore</b>				
	10	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
	9	<b>CS TR Invalid Tile Detection</b>				
	8	<b>VECS Context Switch Interrupt</b>				
	7	<b>Legacy Context Per Process Page Fault Interrupt</b> Fault interrupt is generated by GA fabric, not by the CS This interrupt is for handling Legacy context PPTGTT Page Fault.				
	6	<b>VECS Watchdog Counter Expired</b>				
	5	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
	Format:	MBZ				
4	<b>VECS MI Flush DW Notify</b>					
3	<b>VECS Error Interrupt</b>					
2:1	<b>Reserved</b> <table border="1"> <tr> <td>Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ	
Access:	RO					
Format:	MBZ					
0	<b>VECS MI User Interrupt</b>					



## 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	<b>MbY</b>
		Format: U8
	23:16	<b>MbX</b>
		Format: U8
	15:8	<b>MbClock16</b>
		Format: U8
	7:3	<b>Reserved</b>
		Access: RO
Format: MBZ		
2	<b>MbRcFlag</b>	
	Format: U1	
1	<b>MBLevelInterMBConformanceFlag</b>	
	Format: U1	
0	<b>MBLevelIntraMBConformanceFlag</b>	
	Format: U1	
1	31:29	<b>Reserved</b>
		Access: RO
		Format: MBZ
	28:16	<b>MB_Residual_BitCount</b>
		Format: U13
15:13	<b>Reserved</b>	
	Access: RO	
	Format: MBZ	
12:0	<b>MB_Total_BitCount</b>	
	Format: U13	
2	31:25	<b>Reserved</b>
		Access: RO
		Format: MBZ
	24:0	<b>Cbp</b>



VP8 Encoder StreamOut Format		
		Format: U25
3	31	<b>Reserved</b>
		Access: RO Format: MBZ
	30	<b>LastMbFlag</b>
		Format: U1
	29	<b>IntraMBFlag</b>
		Format: U1
	28:24	<b>MbType5Bits</b>
		Format: U5
	23:19	<b>Reserved</b>
		Access: RO Format: MBZ
	18	<b>QindexClampHigh</b>
		Format: U1
17	<b>QindexClampLow</b>	
	Format: U1	
16	<b>CoeffClampStatus</b>	
	Format: U1	
15:0	<b>Reserved</b>	
	Access: RO Format: MBZ	



## WDBoxOAIInterrupt Vector

WDOA_INTR_VEC - WDBoxOAIInterrupt Vector			
Size (in bits):	16		
Default Value:	0x00000000		
DWord	Bit	Description	
0	15:13	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	12	<b>Performance Monitoring Buffer Half-Full Interrupt</b> For internal trigger (timer event based) reporting, this interrupt is generated if the report buffer crosses the half full limit.	
	11:8	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	7:6	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	5	<b>WDBox 2 Status Interrupt</b>	
	4	<b>WDBox 2 End of Frame Interrupt</b>	
	3:2	<b>Reserved</b>	
		Access:	RO
Format:		MBZ	
1	<b>WDBox 1 Status Interrupt</b>		
0	<b>WDBox 1 End of Frame Interrupt</b>		

## WD Interrupt Bit Definition

WD Interrupt Bit Definition			
Size (in bits):	16		
Default Value:	0x00000000		
The WD Interrupt Registers all share the same bit definitions from this table.			
DWord	Bit	Description	
0	15:8	<b>Reserved</b>	
		Access:	RO
		Format:	MBZ
	7	<b>WD_Frame_Complete</b> This event occurs when WD capture fully completes a frame.	
	6	<b>WD_GTT_Fault</b> This event occurs when a GTT fault is detected.	
	5	<b>WD_Vblank</b> This event occurs at the start of the WD internal vertical blank.	
	4	<b>Unused_Int_4</b> These interrupts are currently unused.	
	3	<b>WD_Capturing</b> This event occurs when WD capture starts to capture pixels.	
	2	<b>WD_Writes_Complete</b> This event occurs when WD capture data writes complete for the current frame, before the data has been flushed to memory. WD Frame Complete should be used to find when the captured data can be accessed.	
	1	<b>Unused_Int_1</b> These interrupts are currently unused.	
0	<b>Unused_Int_0</b> These interrupts are currently unused.		



## Word Data Payload Register

<b>MDCR_W - Word Data Payload Register</b>						
Size (in bits):		256				
Default Value:		0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000				
DWord	Bit	Description				
0	31:16	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
15:0	<b>Word0</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Specifies the slot 0 data in this payload register	Format:	U16			
Format:	U16					
1	31:16	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
15:0	<b>Word1</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Specifies the slot 1 data in this payload register	Format:	U16			
Format:	U16					
2	31:16	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
15:0	<b>Word2</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Specifies the slot 2 data in this payload register	Format:	U16			
Format:	U16					
3	31:16	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
15:0	<b>Word3</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Format:</td> <td style="width: 30%;">U16</td> </tr> </table> Specifies the slot 3 data in this payload register	Format:	U16			
Format:	U16					
4	31:16	<b>Reserved</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Access:</td> <td style="width: 30%;">RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
Access:	RO					
Format:	MBZ					

<b>MDCR_W - Word Data Payload Register</b>						
	15:0	<b>Word4</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>U16</td> </tr> </table> <p>Specifies the slot 4 data in this payload register</p>	Format:	U16		
Format:	U16					
5	31:16	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
	15:0	<b>Word5</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>U16</td> </tr> </table> <p>Specifies the slot 5 data in this payload register</p>	Format:	U16		
Format:	U16					
6	31:16	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
	15:0	<b>Word6</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>U16</td> </tr> </table> <p>Specifies the slot 6 data in this payload register</p>	Format:	U16		
Format:	U16					
7	31:16	<b>Reserved</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Access:</td> <td>RO</td> </tr> <tr> <td>Format:</td> <td>MBZ</td> </tr> </table>	Access:	RO	Format:	MBZ
	Access:	RO				
Format:	MBZ					
	15:0	<b>Word7</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td>U16</td> </tr> </table> <p>Specifies the slot 7 data in this payload register</p>	Format:	U16		
Format:	U16					



## Word SIMD8 Atomic Operation CMPWR Message Data Payload

<b>MDP_AOP8_W2 - Word SIMD8 Atomic Operation CMPWR Message Data Payload</b>		
Size (in bits):	512	
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000	
<b>DWord</b>	<b>Bit</b>	<b>Description</b>
0.0-0.7	255:0	<b>Src0</b>
		Format: <b>MDCR_W</b> Specifies the Slot [7:0] Source 0 data
1.0-1.7	255:0	<b>Src1</b>
		Format: <b>MDCR_W</b> Specifies the Slot [7:0] Source 1 data

## Word SIMD8 Data Payload

MDP_W_SIMD8 - Word SIMD8 Data Payload				
Size (in bits):	256			
Default Value:	0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<b>Data[7:0]</b> <table border="1"> <tr> <td>Format:</td> <td><b>MDCR_W</b></td> </tr> </table> Specifies the Slot [7:0] data	Format:	<b>MDCR_W</b>
Format:	<b>MDCR_W</b>			



## Word SIMD16 Atomic Operation CMPWR Message Data Payload

MDP_AOP16_W2 - Word SIMD16 Atomic Operation CMPWR Message Data Payload				
Size (in bits):	1024			
Default Value:	0x00000000, 0x00000000			
DWord	Bit	Description		
0.0-0.7	255:0	<b>Src0[7:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;"><b>MDCR_W</b></td> </tr> </table> Specifies the Source 0 data for Slot [7:0]	Format:	<b>MDCR_W</b>
Format:	<b>MDCR_W</b>			
1.0-1.7	255:0	<b>Src0[15:8]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;"><b>MDCR_W</b></td> </tr> </table> Specifies the Source 0 data for Slot [15:8]	Format:	<b>MDCR_W</b>
Format:	<b>MDCR_W</b>			
2.0-2.7	255:0	<b>Src1[7:0]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;"><b>MDCR_W</b></td> </tr> </table> Specifies the Source 1 data for Slot [7:0]	Format:	<b>MDCR_W</b>
Format:	<b>MDCR_W</b>			
3.0-3.7	255:0	<b>Src1[15:8]</b> <table border="1" style="width: 100%;"> <tr> <td style="width: 80%;">Format:</td> <td style="text-align: center;"><b>MDCR_W</b></td> </tr> </table> Specifies the Source 1 data for Slot [15:8]	Format:	<b>MDCR_W</b>
Format:	<b>MDCR_W</b>			



## Word SIMD16 Data Payload

<b>MDP_W_SIMD16 - Word SIMD16 Data Payload</b>				
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	<b>Data[7:0]</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_W</b></td> </tr> </table> Specifies the Slot [7:0] data	Format:	<b>MDCR_W</b>
Format:	<b>MDCR_W</b>			
1.0-1.7	255:0	<b>Data[15:8]</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Format:</td> <td style="text-align: center;"><b>MDCR_W</b></td> </tr> </table> Specifies the Slot [15:8] data	Format:	<b>MDCR_W</b>
Format:	<b>MDCR_W</b>			